DETECTION OF DIGITAL SIGNALS TRANSMITTED OVER A KNOWN TIME INVARIANT CHANNEL

by

SER WEE, B.Sc.

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy at the Loughborough University of Technology.

May 1982

Supervisor : A.P. Clark
Department of Electronic and Electrical Engineering

CONTENTS

PAGE

Appendices

ABSTRACT

This thesis investigates various detection processes that operate with known time invariant channels. The investigations are divided into two main areas, the first of which involves uncoded digital signals. Three different detection processes have been studied here and some promising systems have been developed from these. The first of the detection processes is an iterative detection process whereas the second detection process involves the linear filtering of the received signal. Binary signals are considered in the investigations here. The third detection process achieves the near-maximum likelihood detection of a 16-point QAM digital signal transmitted over a telephone circuit at 9600 bits/second. The detector here operates on the received sample values directly without using any complex prefiltering. The second area of investigation covered in this thesis involves coded digital signals. Binary and 16-point QAM signals have been considered here. Rate $\frac{1}{2}$ and $\frac{2}{3}$ non-systematic convolutional codes with optimum free distance have been used in conjunction with the appropriate Gray codes for the encoding of the signals. At the receiving end, a joint near-maximum likelihood detection/decoding process is used. Computer simulation tests have shown that the system improves the tolerance to Gaussian noise over the corresponding uncoded system at low error rates.

ACKNOWLEDGEMENTS

The author would like to express his gratitude to his supervisor, Dr. A.P. Clark, for his considerable help and guidance.

The financial supports of the Singapore government and the department of Electronic/Electrical engineering at the Loughborough University of Technology, which made the work possible, are gratefully acknowledged.

The author would like to thank his family, especially his father, and Miss Low Wai Ming for their constant encouragements.

GLOSSARY OF SYMBOLS AND TERMS

j	when not used as a subscript letter, it is $\sqrt{-1}$

Vectors are treated as row matrices

$\|b\|$	magnitude (absolute value) of a scalar b
$\|\|B\|\|$	length (Euclidean norm) of a vector B
$\{b_i\}$	components of a vector B
$\{B_i\}$	a set of vectors $B_1, B_2, \cdots$
B^T	transpose of a vector or matrix B
B^{-1}	inverse of a nonsingular matrix B
B(z)	z-transform of a set of sample values given by the components of a vector B
Re(b)	real part of complex-valued quantity b
Im(b)	imaginary part of a complex-valued quantity b
$\{s_i\}$	transmitted data symbols whose values are to be detected at the receiver
$\{s_i'\}$	detected values of $\{s_i\}$
$\{x_i\}$	estimates of $\{s_i\}$
$\delta(t)$	unit impulse function (dirac function)

A signal element is a unit component of a digitally coded signal

E' — average transmitted energy per signal element

$y(t)$ — impulse response of the baseband channel

V — $(g+1)$-component vector whose real or complex components are the sample values of $y(t)$

$$V = \begin{bmatrix} y_0 & y_1 & \cdots & y_g \end{bmatrix}$$

$\{r_i\}$ — sample values of the received signal

$\{w_i\}$ — noise components in the received sample values $\{r_i\}$

T — sampling interval

$\frac{1}{2}N_o$ — Two-sided power spectral density of zero mean stationary additive white Gaussian noise added at the input to the receiver filter

P_e — probability of bit error in the detection process

SNR — signal to noise ratio

X — n-component vector whose jth component is x_j (chapter 3)

Y — $n \times n$ matrix whose real or complex components are given by the n vectors $\{Y_i\}$, $i = 1, 2, \cdots, n$, where

$$Y_i = \begin{bmatrix} \overbrace{0 \quad \cdots \quad 0}^{i-1} & \overbrace{y_0 \quad y_1 \quad \cdots \quad y_g}^{g+1} & \overbrace{0 \quad \cdots \quad 0}^{n-i-g} \end{bmatrix}$$

n_c — number of iterative cycles used in the detection process (chapter 3)

n_s — number of sequential operations involved in the detection process (chapter 3)

$\{t_h\}$ — threshold values used in the detection process (chapters 3 and 4)

X_i — n-component vector containing the components x_{i-n+1}, x_{i-n+2}, $\cdots$, x_i (chapters 5 and 7), or

4n-component vector containing the components $x_{4(i-n+1)-3}$, $x_{4(i-n+1)-2}$, $\cdots$, x_{4i} (chapter 8)

m — number of stored vectors $\{X_i\}$ used in the near-maximum likelihood detection process (chapters 5, 7, and 8)

m_v — value of m for the Viterbi-algorithm detection process

C_i — cost of the vector X_i

C_i' — modified value of C_i

f — number of components of V temporarily ignored in systems 2 - 8 (chapter 5)

G — generator matrix of a convolutional code (chapters 6 - 8)

$G^{(e)}$ — eth code generator of a convolutional code (chapters 6 - 8)

m_e — memory of a convolutional code

ν — constraint length of a convolutional code

A set of vectors $\{B_i\}$, $i = 1, 2, \cdots n$, are linearly independent provided that no set of constants $z_1, z_2, \cdots, z_n$ exists (at least one z_i must be non-zero) such that

$$z_1B_1 + z_2B_2 + \cdots + z_nB_n = 0$$

The rank of a matrix B is the number of linearly independent rows of B

CHAPTER 1

INTRODUCTION

1.1 Background of Research Project

Detection processes for distorted digital signals have been widely studied.[B1-B37,C1-C43] The early approach was to use a filter (equalizer) to compensate for the signal distortion introduced by the channel. These equalizers are usually implemented as transversal filters. If the channel introduces a pure phase distortion, then the optimum available tolerance to additive white Gaussian noise can be achieved by a linear equalizer.[A9,B34] Most practical channels, however, introduce some amplitude distortion and under this condition, a better tolerance to noise can always be achieved by a nonlinear equalizer. The weakness of the nonlinear equalizer is that, only a portion of the signal energy is used in the detection of that signal. A better tolerance to additive white Gaussian noise can therefore be achieved by the more sophisticated detection processes that use more of the available signal energy than that used by the nonlinear equalizer.[C1-C43] Many of these detection processes, however, require considerable equipment complexity.

An iterative detection process that involves a sequence of similar operations which are performed successively by a fairly simple piece of equipment has recently been studied.[C5,C28,C29,C30,C32,C36] Although this detection process is more complex than the nonlinear equalizer, it is able to achieve a tolerance to noise near to the optimum available value.[C36]

It has been known that the optimum performance can be achieved by the maximum likelihood (ML) detection process which uses all the available signal energy in the detection of that signal.[A9,A10] However, it is not feasible to implement the ML detection process in practice because of its excessive equipment complexity involved. It has also been known that with sufficient delay in detection, a detection process employing the Viterbi algorithm (VA) can achieve (for practical purposes) the same tolerance to noise as that of the ML detection process.[A9,A10] The Viterbi algorithm was originally used for decoding a convolutional code. Although the VA detector requires far fewer operations and much less storage than the ML detection process, its equipment complexity still grows exponentially with the length of the channel sampled impulse response and with the number of possible signal values. In cases where the duration of the channel sampled impulse response is long and when a multi-level signal is used, the VA detector can become too complex to be used in practice. One approach for overcoming this problem is to use a linear prefilter to shorten the length of the channel sampled impulse response before applying the VA detector to the resultant received signal.[C13,C16,C20,C33] The alternative approach is to modify the VA detection process itself so that the number of operations per received data symbol and the amount of storage are reduced to some managable quantities without degrading the tolerance to noise by a significant amount. This has led to the developments of the near-maximum likelihood detection processes.[C31,C32,C35,C37,C38,C41,C42] When necessary, the linear prefilter used in the former approach

can of course be used here too. Unfortunately, it has not been possible to achieve a satisfactory tolerance to noise when the near-maximum likehood detection process is applied directly (without the use of a complicated linear prefilter) to the very badly distorted 16-point QAM signals mentioned in reference C42.

The aim of the first part of this thesis is to investigate the iterative detection process yet further in an attempt to reduce the equipment complexity and the number of operations required per data symbol. A slightly different detection process, referred to as the systematic search detection process, is also developed and studied. Furthermore, the near-maximum likelihood detection process is also studied here. The object here is to develop a more efficient, and probably therefore more complex system that is able to achieve a satisfactory tolerance to noise with a severely distorted 16-point QAM signal level, without the use of a complicated linear prefilter.

For a coded system, an error correcting code is used to reduce the error rate. Block codes were first used for this purpose until the invention of convolutional codes, which are easier to implement in practice and are generally more efficient.(D1-D49) The conventional system carries out the detection and decoding processes separately. Recently, convoluticnal codes have been used with a joint detection/ decoding process, which appear to have a better tolerance to noise than that of the conventional system.(D39,D40) The drawback of these coded systems is that the information rate is reduced due to the added redundant coded digits. Nevertheless, it has also been known

that some multi-level coding schemes can be used in systems operating over non-dispersive channels, and the coding process now does not reduce the information rate.

The aim of the second part of this thesis is to investigate the arrangement where a multi-level convolutional coding scheme and a joint detection/decoding process are used in a system operating over a dispersive channel.

1.2 Outline of Topics Studied

The research project is concerned with the investigation of the various detection processes for use in a synchronous serial data transmission system. The aim is to obtain a better understanding of these detection processes and to develop the most cost effective arrangements for the different applications. Since these detection processes are highly nonlinear, an exact theoretical quantitative analysis of their performances is extremely complicated and difficult, if not impossible, with the present state of the art. Furthermore, these detection processes all involve the processing of sets of numerical values. The performances of these computer-like detection processes are therefore best evaluated by using computer simulation. This is the approach that is used most often here. Nevertheless, some qualitative analysis of the systems studied and developed here have also been carried out, where possible, in the investigations. The channel sampled impulse response is assumed to be known and time-invariant so that some computation using this prior knowledge can always be carried out at the receiver before the transmission

begins. The research project has been divided into two areas, the first of which deals with the uncoded systems and is discussed in chapters 2,3,4, and 5, whereas the second area deals with the coded systems and is discussed in chapters 6,7, and 8.

Chapter 2 gives the mathematical model of the synchronous serial data transmission system for the uncoded systems to be studied and developed in chapters 3,4, and 5. Various designs of the conventional linear and nonlinear equalizers, and the maximum likelihood detection process are also described in this chapter.

In chapter 3, various iterative detection processes, to be used with binary baseband signals, are studied and developed.

In chapter 4, a systematic search detection process, to be used with binary baseband signals, is studied and developed.

In chapter 5, near-maximum likelihood detection processes for a 16-point QAM signal transmitted over a telephone circuit are further investigated. Systems with a simple linear prefilter at the detector input are first studied and tested. Other systems which discard entirely the use of any linear prefilter are then developed and tested.

While chapter 2 gives the model of the data transmission system for the uncoded systems, chapter 6 gives the model of the data transmission system for the coded systems to be studied and developed in chapters 7 and 8. The structure of the convolutional encoder and the Viterbi decoding technique are also briefly described here.

Chapter 7 is concerned with the transmission of binary signals in the coded systems. Two coding schemes, both involving the use of a rate-$\frac{1}{2}$ convolutional code and a Gray code, have been used here to convert the binary signal into a quaternary signal for transmission.

In chapter 8, the more promising coding scheme in chapter 7 is applied to some telephone circuits employing 16-point QAM signals. Three coding schemes, which involve the use of a rate-$\frac{1}{2}$ or rate-$\frac{2}{3}$ convolutional code and the appropriate Gray code, are used here to convert the 16-point QAM signal into the 256-point or the 64-point QAM signal for transmission.

Computer simulation tests are used in this thesis to determine the tolerances to noise of the various systems studied and developed, and the results are plotted as graphs of bit error rate versus signal to noise ratio. In order to avoid using excessive computer time, the bit error rates evaluated here are limited to be above 10^{-4}. The number of data symbols transmitted in any one measurement of the bit error rate is 10,000. Thus, for a system employing the 16-point QAM signals, the total number of bits transmitted becomes 40,000 in any one measurement of the bit error rate since each data symbol here contains 4 bits. Nevertheless, when the simulation results are more scattered which normally occurs at the lower bit error rates, several bit error rate measurements are carried out using the same signal to noise ratio.

For convenience, vectors are also treated as row matrices in this thesis.

CHAPTER 2

SYSTEMS WITH UNCODED SIGNALS

2.1 Model of the Data-Transmission System for Binary Signals

The data transmission system studied in chapters 3 and 4 is the synchronous serial binary baseband data transmission system shown in Fig. 2.1.

The information to be transmitted is carried by the data symbols $\{s_i\}$ and is fed to the input of the transmitter filter as a stream of regularly spaced impulses $\{s_i\delta(t-iT)\}$. The $\{s_i\}$ are assumed to be statistically independent and equally likely to have any of their 2 possible binary values, or

$$s_i = \pm 1 \qquad (2.1)$$

In practice, a rectangular or rounded waveform would of course be used instead of an impulse, with the appropriate change in the transmitter filter.

The transmission path could either be a lowpass channel, with a frequency limit no greater than about 10kHz, or a typical voice frequency channel with a frequency band no wider than about 3000 Hz, such as a telephone network or a H.F. radio link. In the latter case, the transmission path in Fig. 2.1 is assumed to include a linear modulator (at the transmitter) and a linear demodulator (at the receiver), so that the whole of the transmitter filter, the transmission path, and the receiver filter can be considered as a linear baseband channel.(A8,A9)

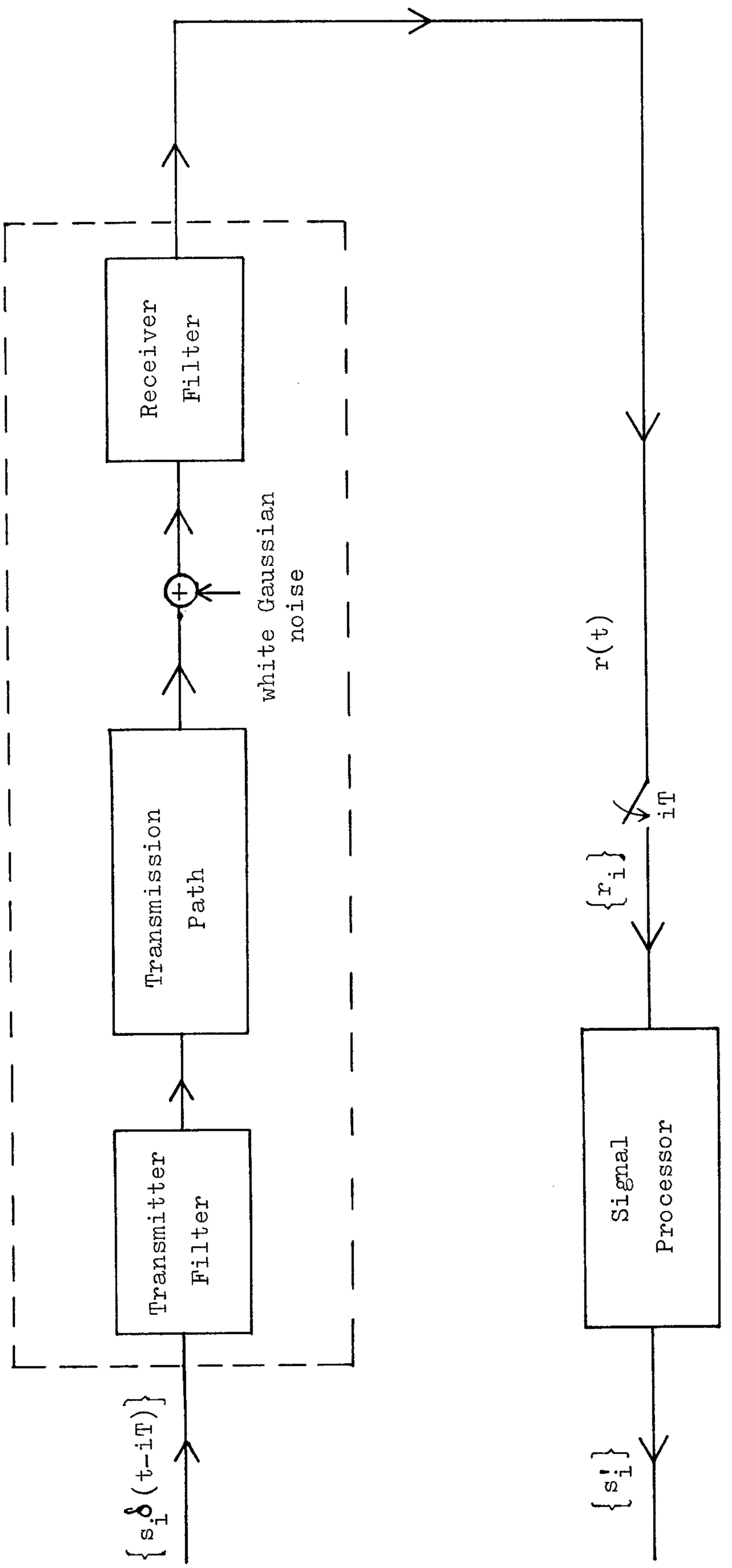


Fig. 2.1 Synchronous serial binary baseband data-transmission system

The impulse response y(t) of the linear baseband channel in Fig. 2.1 has, for practical purposes, a finite duration of less than (g+1)T seconds, where g is a positive integer and T seconds is the time interval between adjacent impulses at the input to the transmitter. It is assumed that y(t) is known and time invariant during the whole transmission period.

The transmitter filter shapes the transmitted signal to the approximate bandwidth of the transmission path, while the receiver filter removes the noise outside the frequency band of the received signal.

Additive noise is the most important type of noise introduced by the practical channels such as the telephone circuits or the voice frequency channels using H.F. radio link.[A8] It has been shown that if a data-transmission system has a better tolerance to additive white Gaussian noise than another, it will, in general, also have a better tolerance to other types of additive noise.[A8] Furthermore, the effects of additive white Gaussian noise on a digital data-transmission system may readily be analysed theoretically and atudied by computer simulation. It is for these reasons that the noise introduced at the output of the transmission path in Fig. 2.1 is assumed to be stationary additive white Gaussian noise. This noise has zero mean and a two-sided power spectral density of $\frac{1}{2}N_o$. It should be mentioned that, although the use of additive white Gaussian noise is very convenient for the mathematical model of the data-transmission system studied here, it is not physically realisable since a white noise has an infinite bandwidth and hence an infinite power level for a non-zero power spectral density.

The signal at the output of the receiver **filter** in Fig. 2.1 is

$$r(t) = \sum_i s_i y(t-iT) + w(t) \qquad (2.2)$$

where $w(t)$ is the zero mean Gaussian noise waveform at the output of the receiver filter. The waveform $r(t)$ is sampled once per signal element at the time instants $\{iT\}$, where i takes on all positive integer values. The sampling instants are assumed to be correctly synchronised to the received signal.

The sampled impulse response of the baseband channel in Fig. 2.1 is given by the $(g+1)$-component vector

$$V = \left[y_0 \; y_1 \; \cdot\cdot \; y_g \right] \qquad (2.3)$$

where $y_i = y(iT)$. The delay in detection, other than that involved in the time dispersion of the transmitted signal, is neglected here, so that $y_0 \neq 0$ and $y_i = 0$ for $i < 0$ and $i > g$.

Thus, the sample value of the received signal at the output of the receiver filter, at time $t = iT$, is

$$r_i = \sum_{k=0}^{g} s_{i-k} y_k + w_i \qquad (2.4)$$

where $r_i = r(iT)$ and $w_i = w(iT)$.

It is assumed that the receiver filter is such that the noise components $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and variance $\sigma^2 = \frac{1}{2}N_o$. (A9)

The signal processor in Fig. 2.1 operates on the $\{r_i\}$ to give the detected data symbol values designated as $\{s_i'\}$. It is assumed that the signal processor has prior knowledge of both the channel

sampled impulse response V and the two possible values of the data symbol s_i.

In all practical data-transmission systems, the transmitted signal energy is limited. Consequently, the system must be designed to give the maximum tolorance to noise subject to a transmitted power constraint. In the data-transmission system studied here, the transmitter filter is assumed to be such that the average transmitted energy per data symbol is the mean-square value of s_i and is equal to unity, that is,

$$E\left[s_i^2\right] = 1 \qquad (2.5)$$

since the $\{s_i\}$ are assumed to be statistically independent and equally likely to have any of their two possible values ± 1.

2.2 Signal Distortion Introduced by a Channel Sampled Impulse Response

The signal distortion introduced by a channel sampled impulse response has been described in detail in Ref. A9, and the more important results are quoted here so that they can be used to analyse some of the system performance in the later chapters of this thesis.

Two types of distortion can be introduced by the sampled impulse response of a channel and these are the amplitude and phase distortion which are defined in terms of the DFT (Discrete Fourier Transform) components of the corresponding sampled impulse response.

2.3.1 Phase distortion

The signal distortion introduced by a channel is defined to be pure phase distortion (that is, no amplitude distortion or attenuation) when all the DFT components of the sampled impulse response of this channel have magnitude unity. To see the time domain equivalence of this definition, let V(z) be the z-transform of the channel sampled impulse response so that

$$V(z) = y_0 + y_1 z^{-1} + \cdots + y_g z^{-g} \qquad (2.6)$$

where $y_0, y_1, \cdots, y_g$ are all real-valued quantities. Furthermore, let U(z) be the z-transform of the sequence of values obtained by reversing the order of the sequence with z-transform V(z), the reversal being pivoted about the component at time t = 0. Thus,

$$U(z) = y_0 + y_1 z^{+1} + \cdots + y_g z^{+g} \qquad (2.7)$$

It is shown in Ref. A9 that a channel with z-transform V(z) (eqn. 2.6) introduces pure phase distortion when the sequence with z-transform U(z) (eqn. 2.7) is the same as the sequence with z-transform $V^{-1}(z)$. That is, if the reversed sequence of a channel sampled impulse response is also the inverse sequence of it, then the signal distortion introduced by this channel is a pure phase distortion.

Some properties of the aperiodic autocorrelation function of the channel sampled impulse response introducing pure phase distortion are now described. The z-transform of the aperiodic autocorrelation function of the channel sampled impulse response is

$$V(z)\ z^{-g}\ U(z) = \bar{a}_0 + \bar{a}_1 z^{-1} + \cdots + \bar{a}_{2g} z^{-2g} \qquad (2.8)$$

where
$$\bar{a}_i = \sum_{h=-\infty}^{\infty} y_h y_{i-g+h} \qquad (2.9)$$

for $i = 0,1, \cdots, 2g$ and $y_h = 0$ for $h < 0$, $h > g$. $V(z)$ and $U(z)$ are, of course, as defined by eqns. 2.6 and 2.7. It can be seen from eqn. 2.9 that

$$\bar{a}_{g-i} = \bar{a}_{g+i} \qquad (2.10)$$

for $i = 1,2, \cdots, g$. For a channel introducing pure phase distortion, its aperiodic autocorrelation function has been shown to be such that[A9]

$$\begin{aligned} \bar{a}_i &= 0, \quad \text{when } i \neq g \\ &= 1, \quad \text{when } i = g \end{aligned} \qquad (2.11)$$

where the $\{\bar{a}_i\}$ are as defined by eqns. 2.8 and 2.9. This means that the aperiodic autocorrelation function of the channel sampled impulse response is unaffected by channel phase distortion.

If s_i is the data symbol carried by the ith transmitted signal element, then the components of the corresponding individual received signal element at the output of the sample in Fig. 2.1 are $s_i y_0$, $s_i y_1$, $\cdots$, $s_i y_g$ which are spread over (g+1) sampling intervals of time. Similarly, the components of another individual received signal element associated with s_{i+h} are $s_{i+h} y_0$, $s_{i+h} y_1$, $\cdots$, $s_{i+h} y_g$ where h takes on any integer value not equal to zero. Clearly, when h is a positive integer, each component of the sequence of values $s_{i+h} y_0$, $s_{i+h} y_1$, $\cdots$, $s_{i+h} y_g$ is received at a delay of h sampling intervals after the receipt of the corresponding component of the sequence of values $s_i y_0$, $s_i y_1$, $\cdots$, $s_i y_g$, and when h is a negative integer, the situation is

reversed. Consequently, it can be seen from eqns. 2.9 and 2.11 that the received signal elements at the output of the sampler in Fig. 2.1 are orthogonal when the channel introduces pure phase distortion. Moreover, the transmitted signal elements in Fig. 2.1 are also orthogonal. All these therefore suggest that a baseband channel that introduces pure phase distortion introduces an orthogonal transformation onto the transmitted signal. Consequently, as Ref. A9 has shown, there is no inevitable loss in tolerance to noise when the received signals $\{r_i\}$ in Fig. 2.1 are processed by the appropriate signal processor.

2.2.2 Amplitude distortion

The signal distortion introduced by a channel is defined to be pure amplitude distortion (that is, no phase distortion or delay) when all the DFT components of the sampled impulse response of this channel are real-valued quantities. The time domain equivalence of this definition is that, the components of the channel sampled impulse response $y_0, y_1, \cdots, y_g$, are symmetrical about its central component $y_{g/2}$ where g is an even value and $y_0, y_1, \cdots, y_g$, are all real-valued quantities. Thus,

$$y_i = y_{g-i} \qquad (2.12)$$

for $i = 0,1, \cdots, ((g/2)-1)$.

It can be shown that a channel introducing pure amplitude distortion does not introduce an orthogonal transformation onto the transmitted signal, and it normally reduces the best tolerance to additive white Gaussian noise regardless of the type of signal processor used at the receiver.(A9) Consequently, amplitude distortion

is a much more important factor (than phase distortion) to be considered when assessing the severity of the signal distortion introduced by a channel.

Since the aperiodic autocorrelation function $\bar{a}_0, \bar{a}_1, \cdots, \bar{a}_{2g}$ depends not on the phase distortion but on the amplitude distortion, and since it is readily derived from the channel sampled impulse response, it can be used as a measure of the severity of the amplitude distortion. Ref. A9 uses the distortion figure defined as

$$\bar{d} = (\bar{a}_g)^{-2} \sum_{i=0}^{g-1} (\bar{a}_i)^2 \qquad (2.13)$$

as a measure of the severity of the amplitude distortion. It is suggested that when $\bar{d} > 0.1$, there is significant amplitude distortion and when $\bar{d} > 0.5$, the distortion is severe.[A9] It should be noticed that $\bar{d} = 0$ when the channel introduces pure phase distortion (eqn. 2.11).

2.2.3 Effects of the locations of zeros of the z-transform of the channel sampled impulse response

Clearly, the use of the distortion figure $\bar{d}$ (eqn. 2.13) in assessing the severity of the amplitude distortion takes no account of the detailed structure or shape of the signal distortion, and two different channels having the same value of $\bar{d}$ can therefore have different effects on the tolerance to noise of a data-transmission system.[A9,C36] It has been found that the locations of zeros of the z-transform $V(z)$ of the channel sampled impulse response in the z-plane determine, to some extent, the severity of the distortion introduced by the baseband channel.[A9,C36] Generally, the distortion is likely to be very severe when the zeros are located on or very near to the

unit circle. The distortion is also likely to be less severe when the zeros of V(z) are located inside the unit circle. The locations of these zeros are related to the distribution of the magnitudes of the components in the channel sampled impulse response. As a rule of thumb, if the large components (in magnitudes) of the channel sampled impulse response are concentrated towards the front, then the zeros are normally located on or inside the unit circle in the z-plane, whereas if the large components of the channel sampled impulse response are concentrated near the end, then the zeros are normally located on or outside the unit circle.

Thus, in assessing the signal distortion introduced by the channel, the distortion figure $\bar{d}$ (eqn. 2.13) may be used when all components of the channel sampled impulse response are real-valued quantities. The locations of the zeros of the channel sampled impulse response, however, may be used to assess the signal distortion introduced by this channel whose components may either be real-valued or complex-valued quantities. The model of the data-transmission system operating over a complex-valued channel sampled impulse response is described in section 2.4.

2.3 Channel Sampled Impulse Responses to be used for the Computer Simulation Tests in Chapters 3 and 4

Three channels are used for the computer simulation tests in chapters 3 and 4 where binary signals are considered. The sampled impulse responses of these three channels, together with their corresponding zeros and distortion figures (eqn. 2.13) are given in Table 2.1. In particular, channel A introduces very severe amplitude

distortion and no phase distortion while channel B being a mild channel, introduces a combination of amplitude and phase distortion. Channel C has a sampled impulse response with many components, whose magnitudes increase slowly from the start until the peak is reached. Furthermore, all zeros of the z-transform of channel C are located on the unit circle in the z-plane as can be seen in Table 2.1. This implies that the distortion introduced by channel C is likely to be very severe.

The sampled impulse response of all these channels are normalised so that,

$$\sum_{h=0}^{g} (y_h)^2 = 1 \qquad (2.14)$$

Channels	Sampled impulse responses V	Zeros of V(z)			Distortion figures (eqn. 2.13)
		Real part	Imaginary part	Magnitude	
A	0.167 0.471 0.707 0.471 0.167	-0.928	-1.031	1.387	0.91
		-0.928	+1.031	1.387	
		-0.482	-0.536	0.721	
		-0.482	+0.536	0.721	
B	0.548 0.789 0.273 -.044 0.012	-0.806	-0.310	0.863	0.42
		-0.806	+0.310	0.863	
		+0.086	-0.149	0.172	
		+0.086	+0.149	0.172	
C	0.049 0.178 0.338 0.467 0.516	-0.316	-0.949	1.000	1.90
	0.467 0.338 0.178 0.049	-0.316	+0.949	1.000	
		+0.309	-0.951	1.000	
		+0.309	+0.951	1.000	
		-0.809	-0.588	1.000	
		-0.809	+0.588	1.000	
		-1.000	0.000	1.000	
		-1.000	0.000	1.000	

Table 2.1 Sampled impulse responses of channels A, B, and C (chapters 3 and 4)

2.4 Model of the Data-Transmission System using 16-point QAM Signals

This is the model of the data-transmission system studied in chapter 5 where various detection processes are developed to operate at 9600 bit/s over the public switched telephone network. The model is basically the same as that described in section 2.1, where a binary signal is used, but with some alterations. Fig. 2.2 shows the block diagram of this model of the data-transmission system.

It can be seen from Fig. 2.2 that a QAM system involves the transmission of two parallel signals each requiring a separate amplitude modulator at the transmitter and a separate amplitude demodulator at the receiver. The two signal carriers have the same frequency but are in phase quadrature (at 90°).

Thus, at the transmitting end, a binary information sequence is fed to the input of the buffer store which holds $\log_2(16)$ or 4 successive binary digits for a 16-point QAM signal. These binary digits are coded into two multi-level signals, $a_i\delta(t-iT)$ and $b_i\delta(t-iT)$, which are then shaped to the appropriate bandwidth before modulating the 'in-phase' and 'quadrature' carriers represented by $\sqrt{2}\cos(2\pi f_c t)$ and $-\sqrt{2}\sin(2\pi f_c t)$ respectively. f_c is here the nominal carrier frequency. The resulting QAM signal is then transmitted over the transmission path. The additive noise is assumed to be white Gaussian noise. At the receiving end, a bandpass filter is used to suppress the out-of-band noise. Coherent demodulation is used and the demodulator is assumed to know the nominal carrier frequency, but has no knowledge of the carrier phase. The phase error $\emptyset$ in the receiver local oscillator may thus have any value

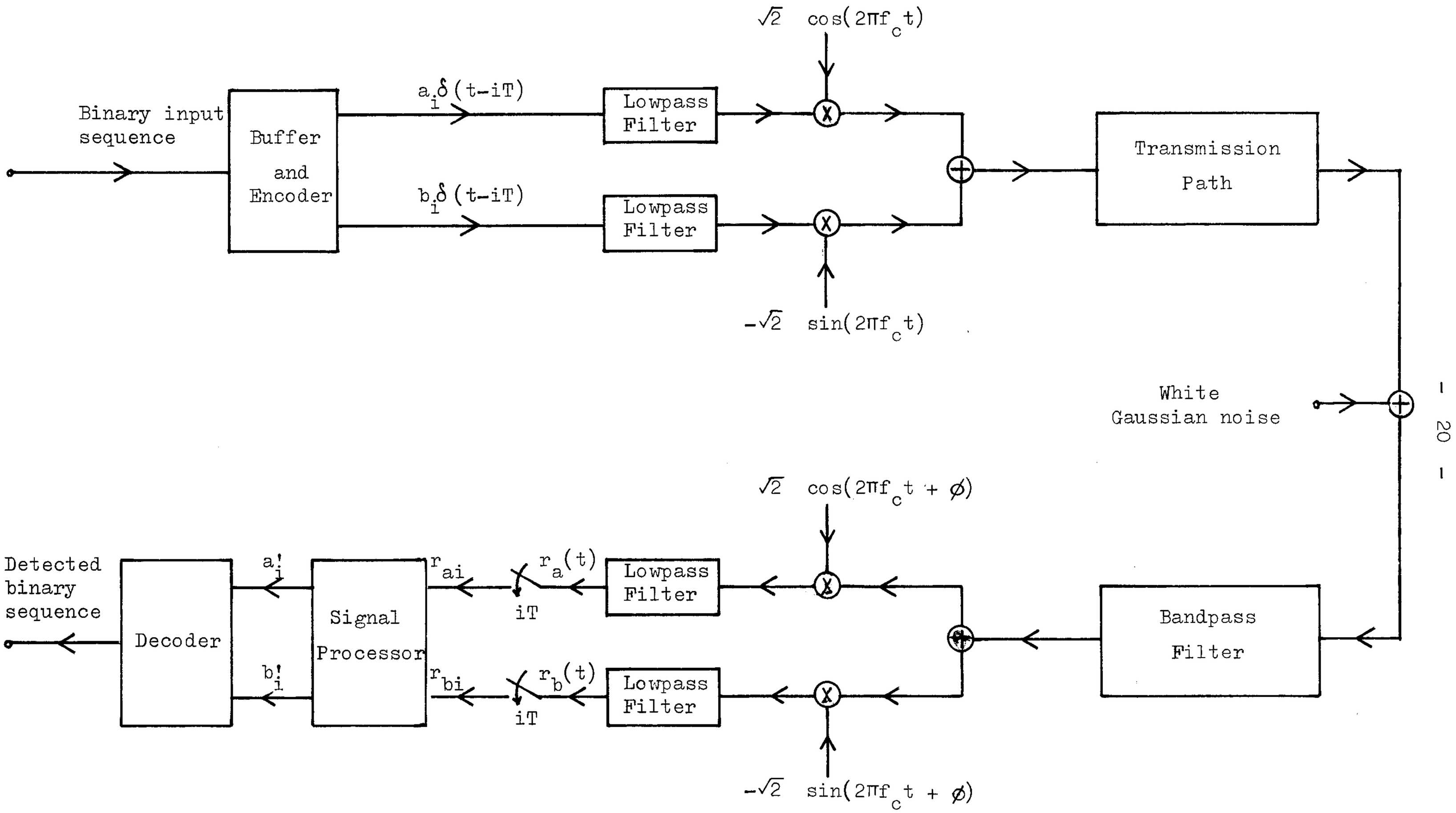


Fig. 2.2 Model of a data-transmission system using QAM signals

between $-\pi$ and $+\pi$ radians. The lowpass filters after the demodulator, suppress the high frequency components so that only the baseband signals are retained. The signal processor then operates on the received in-phase and quadrature sample values to give the detected data symbol values and hence the corresponding decoded binary information sequence.

The model described in Fig. 2.2 can be reduced to an equivalent baseband model with complex signals and complex channel sampled impulse response.[C36] Thus, the linear modulator, the transmission path and the linear demodulator are combined to become a baseband transmission path carrying signals with complex values. The signals transmitted over the in-phase channel (that associated with $\cos(2\pi f_c t)$) are represented by real-valued quantities, and the signals over the quadrature channel (that associated with $\sin(2\pi f_c t)$) by imaginary-valued quantities, to give a resultant complex-valued baseband signal at both the input and output of the baseband transmission path. This equivalent baseband model is shown in Fig. 2.3.

As Fig. 2.3 shows, the information to be transmitted is carried by the complex-valued data symbols $\{s_i\}$ where

$$s_i = a_i + j\, b_i \qquad (2.15)$$

and $j = \sqrt{-1}$. For a 16-point QAM signal, which is the signal considered in chapter 5, the possible values of a_i and b_i are,

$$a_i = \pm 1 \ , \ \pm 3$$

and

$$b_i = \pm 1 \ , \ \pm 3 \qquad (2.16)$$

Fig. 2.4 shows the two-dimensional representation of a 16-point QAM signal.

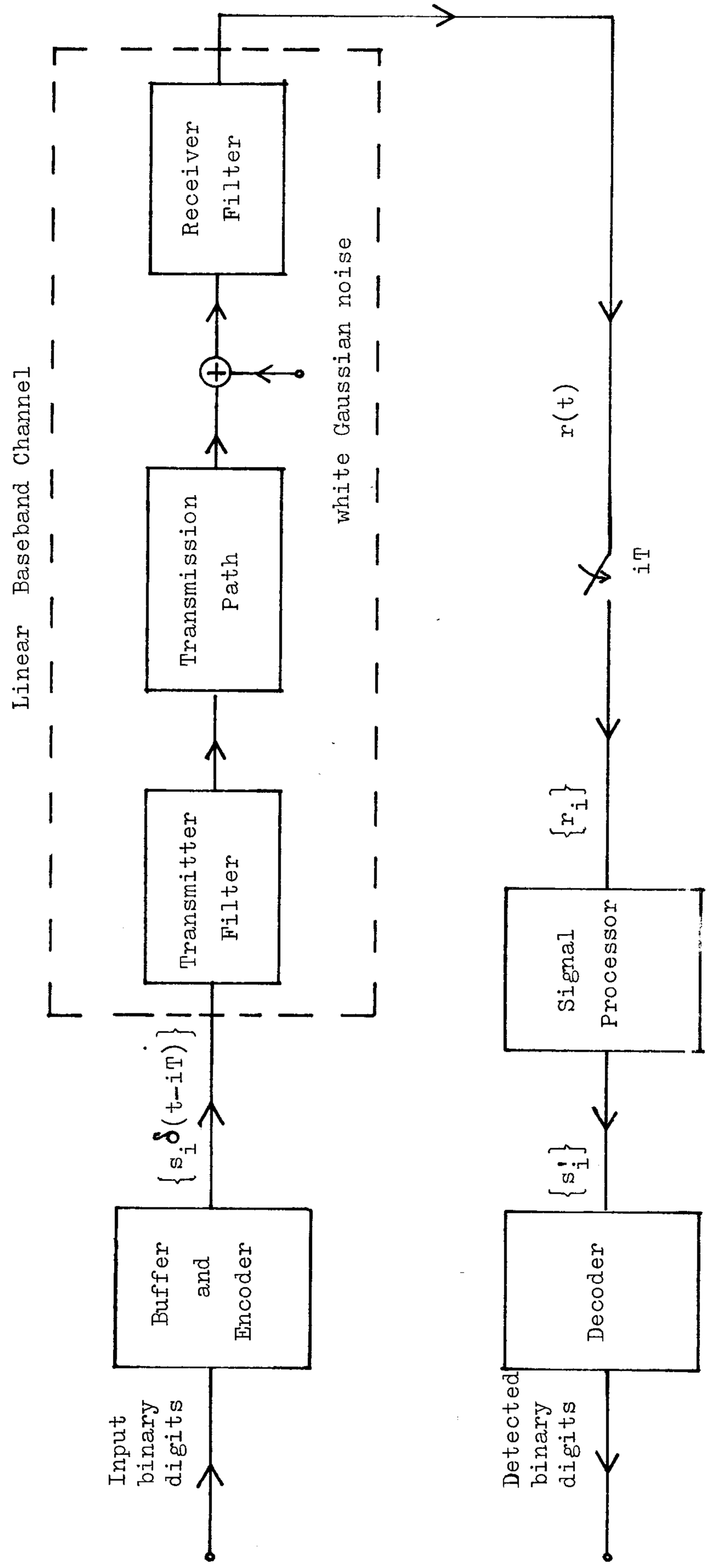


Fig. 2.3 Equivalent baseband model for a QAM data-transmission system

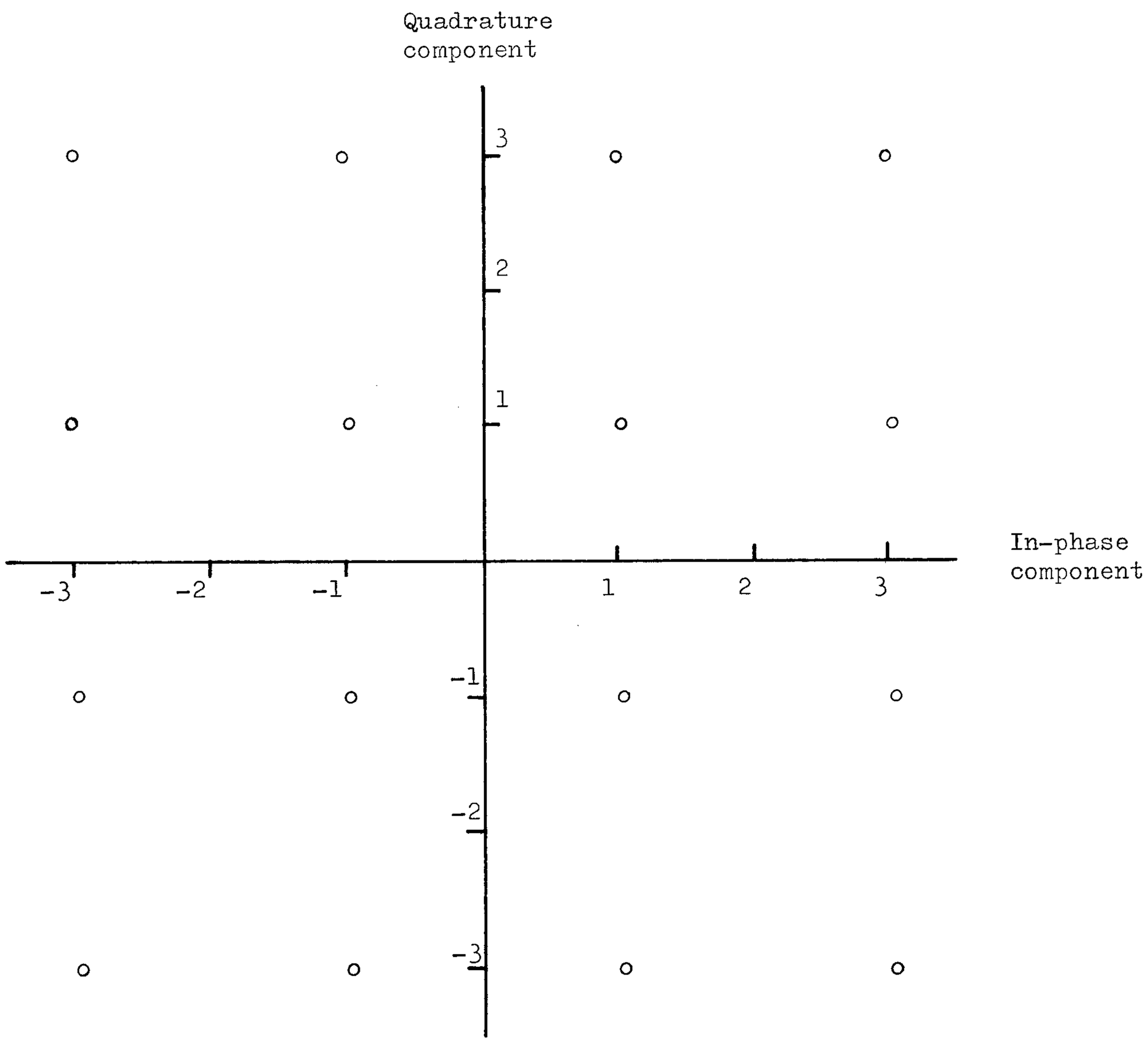


Fig. 2.4 A 16-point QAM signal constellation

The $\{s_i\}$ are statistically independent and equally likely to have any of their 16 possible values. In the particular application considered in this thesis, the transmission path in Fig. 2.3 includes a telephone circuit together with a linear QAM modulator at the transmitter and a linear QAM demodulator at the receiver. The nominal carrier frequency f_c is here equal to 1800 Hz. The transmitter filter, transmission path and receiver filter in Fig. 2.3, together form a linear baseband channel whose impulse response y(t) has for practical purposes, a finite duration of less than (g+1)T seconds, where g is a positive integer and T seconds is the time interval between adjacent impulses at the input to the transmitter. It is assumed in this thesis that y(t) is time-invariant over any one transmission. The value of y(t), at any given value of t, is normally complex.

Stationary white Gaussian noise with complex values, zero mean and a two-sided power spectral density of N_o (twice that of the white Gaussian noise in Fig. 2.2)[C36] is assumed to be the only noise and is added to the data signal at the output of the baseband transmission path, to give the complex-valued Gaussian noise waveform w(t) at the output of the receiver filter. Although telephone circuits do not normally introduce significant levels of Gaussian noise, the relative tolerance of different data-transmission systems to white Gaussian noise is a good measure of their relative overall tolerance to the additive noise actually experienced over telephone circuits.[A8]

Thus, the signal waveform at the output of the receiver filter is

$$r(t) = \sum_i s_i y(t-iT) + w(t) \qquad (2.17)$$

and this is sampled once per signal element (at the Nyquist rate) at the time instants $\{iT\}$, where i takes on all positive integer values. The sampled impulse response of the baseband channel in Fig. 2.3 is given by the (g+1)-component vector

$$V = \left[y_0 \; y_1 \; \cdot\cdot \; y_g \right] \tag{2.18}$$

where $y_i = y(iT)$. The delay in transmission, other than that involved in the time dispersion of the transmitted signal, is neglected here, so that $y_0 \neq 0$ and $y_i = 0$ for $i < 0$ and $i > g$.

The sample value of the received signal at the output of the receiver filter at time t = iT is therefore

$$r_i = \sum_{h=0}^{g} s_{i-h} y_h + w_i \tag{2.19}$$

where $r_i = r(iT)$ and $w_i = w(iT)$. The sample values $\{r_i\}$, $\{s_i\}$, $\{y_i\}$ and $\{w_i\}$ are all complex-valued quantities here.

The signal processor in Fig. 2.3 operates on the $\{r_i\}$ to give the detected data-symbol values $\{s_i'\}$ and hence the corresponding decoded binary information sequence.

The encoder in Fig. 2.3 converts 4 adjacent binary digits of the binary input information sequence into two 4-level data symbols a_i and b_i. In chapter 5, this is done by coding the first two digits into a_i, and the next two digits into b_i using Gray code. The relationship between a_i and its two associated binary digits is the same as that between b_i and its two associated binary digits, and is shown in Table 2.2. These binary input digits are assumed to be statistically independent and equally likely to have any of the values of 0 and 1.

Since a_i and b_i are formed from these binary input digits according to Table 2.2, the complex-valued data symbols $\{s_i\}$ (eqn. 2.15) must also be statistically independent and equally likely to have any of their 16 possible values as is assumed before.

The transmitter filter in Fig. 2.3 is assumed to be such that the average transmitted energy per data symbol is the mean-square value of s_i, which is equal to 10 here as can be seen from eqns. 2.15 and 2.16. The average transmitted energy per bit is therefore 2.5 because s_i, being a 16-point QAM data symbol, carries 4 bits of information.

The receiver filter is also assumed to be such that the real and imaginary parts of the noise components $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and variance $\sigma^2 = \frac{1}{2}N_o$, where N_o is the two-sided power spectral density of the noise added at the output of the transmission path in Fig. 2.3.

The signal processor here (Fig. 2.3) is assumed to have prior knowledge of the channel sampled impulse response, the 16 possible values of s_i, and the Gray code in Table 2.2.

Input binary digits	a_i or b_i
0 0	-3
0 1	-1
1 1	1
1 0	3

Table 2.2 Gray code for s_i (eqn. 2.15)

2.5 Channel Sampled Impulse Responses to be used for the Computer Simulation Tests in Chapter 5

Six baseband channels are used for the computer simulation tests in chapter 5 where 16-point QAM signals are used. The complex-valued channel sampled impulse responses are given in Table 2.4 and Table 2.5. Figs. 2.5 - 2.10 show the locations of the zeros of these channels. The attenuation and group-delay characteristics of the six telephone circuits are shown in Figs. 2.11 - 2.16, the telephone circuits forming a part of the transmission path in Fig. 2.3. Fig. 2.17 shows the resulting attenuation and group-delay characteristics of the equipment filters. The equipment filters are here considered as operating on the transmitted bandpass signals, rather than on the baseband signals in the transmitter and receiver as shown in Fig. 2.3, so as to show more clearly the effects of the equipment filters in limiting some of the distortions introduced by the telephone circuits. The equipment filters are also taken to include the filtering required to convert the sequence of impulses at the transmitter input into the corresponding rectangular waveform used in an actual data-transmission system.

Figs. 2.13 and 2.17 show that nearly all the distortion introduced by channel C is in fact introduced by the equipment filters and not the telephone circuit. It can be seen from Figs. 2.11 - 2.17 that channel A introduces very severe group-delay distortion as well as severe attenuation distortion, channel B introduces very severe attenuation distortion and channels C, D, E, and F all introduce typical levels of both attenuation and group-delay distortions.

Furthermore, Tables 2.6 and 2.7 give the six resultant sampled impulse responses obtained by replacing all the zeros of the corresponding z-transform of channels A-F (Tables 2.4 and 2.5) that lie outside the unit circle in the z-plane, by the complex conjugates of their reciprocals. All zeros of the z-transform of these six resultant sampled impulse responses therefore lie inside the unit circle in the z-plane. The sampled impulse responses shown in Tables 2.4 - 2.7 are normalised so that

$$\sum_{h=0}^{g} y_h y_h^* = \sum_{h=0}^{g} |y_h|^2 = 1 \qquad (2.20)$$

where y_h^* is the complex conjugate of y_h, and $|y_h|$ is the modulus or magnitude of y_h.

The sampled impulse responses given in Tables 2.6 and 2.7 will be further discussed and used in the later parts of this thesis.

Channel A		Channel B		Channel C	
Real part	Imaginary part	Real part	Imaginary part	Real part	Imaginary part
0.0176	-.0175	-.0038	-.0049	0.0326	-.0045
0.1381	-.1252	0.0077	-.0044	0.5483	-.0255
0.4547	-.1885	0.0094	0.0207	0.8031	0.0659
0.5078	0.1622	-.0884	0.0355	-.2430	-.0286
-.1966	0.3505	-.1138	-.2869	0.0066	-.0176
-.2223	-.2276	0.5546	-.2255	0.0307	0.0180
0.2797	-.0158	0.1903	0.5813	-.0170	-.0115
-.1636	0.1352	-.2861	-.0892	0.0052	0.0056
0.0594	-.1400	0.2332	-.0384	-.0041	-.0028
-.0084	0.1111	-.0652	0.0428	0.0021	0.0017
-.0105	-.0817	0.0335	-.0519	-.0001	0.0001
0.0152	0.0572	-.0323	0.0170	-.0017	-.0004
-.0131	-.0406	0.0044	-.0023	0.0010	-.0002
0.0060	0.0255	0.0054	0.0076	0.0006	0.0001
0.0003	-.0190	0.0008	-.0051	-.0013	0.0000
-.0035	0.0116	-.0056	0.0001	0.0004	0.0002
0.0041	-.0078	0.0018	0.0032	0.0004	0.0000
-.0031	0.0038	-.0009	-.0015	-.0002	0.0001
0.0018	-.0005	-.0022	-.0026	0.0001	-.0004
-.0018	-.0005	0.0029	0.0019	-.0005	0.0003
0.0007	0.0007	-.0008	0.0009	0.0005	0.0000
0.0004	0.0001	-.0014	-.0003	0.0001	0.0000
-.0004	0.0001	0.0019	-.0002	-.0003	0.0001
-.0001	0.0010	-.0003	0.0005	0.0001	-.0001
0.0000	-.0007	0.0007	0.0005	0.0003	0.0002
0.0004	0.0008	-.0007	-.0001	-.0001	-.0002
-.0002	0.0000	0.0002	-.0008	-.0003	0.0001
0.0000	-.0004	0.0006	0.0000	0.0003	0.0001
0.0002	-.0002	0.0002	0.0004	0.0001	-.0001
0.0000	-.0001	-.0001	-.0004	-.0002	0.0001
-.0001	-.0001			0.0000	-.0001
-.0001	-.0002			0.0002	0.0001
0.0004	-.0003			-.0001	-.0001
-.0002	0.0003			-.0001	0.0000
-.0002	-.0002			0.0001	0.0001
0.0000	0.0000			0.0000	-.0001
0.0001	0.0000			0.0001	0.0001
0.0001	0.0002			-.0001	-.0001
-.0001	0.0003			0.0000	0.0000
-.0001	0.0000			0.0002	0.0000
0.0002	0.0000			-.0001	0.0000
-.0001	0.0002			0.0000	0.0000
0.0000	-.0001			0.0001	0.0000
0.0001	0.0000			-.0001	0.0000
0.0000	0.0001			-.0001	0.0000

Table 2.4 Sampled impulse responses of channels A, B, and C. (chapter 5)

Channel D		Channel E		Channel F	
Real part	Imaginary part	Real part	Imaginary part	Real part	Imaginary part
0.0145	-.0006	-.0086	0.0030	-.0291	-.0373
0.0750	0.0176	0.0004	0.0042	0.2290	-.2244
0.3951	0.0033	0.0059	-.0094	0.7612	0.1817
0.7491	-.1718	-.0409	-.0090	0.2988	0.3050
0.1951	0.0972	0.0043	0.0405	-.0338	-.2915
-.2856	0.1894	0.4725	0.0186	-.0789	0.0616
0.0575	-.2096	0.8081	-.0218	0.0291	0.0287
0.0655	0.1139	0.0105	-.0010	-.0137	-.0352
-.0825	-.0424	-.1972	-.0450	0.0020	0.0204
0.0623	0.0085	0.2039	0.0602	0.0004	-.0108
-.0438	0.0034	-.1028	-.0803	0.0028	0.0065
0.0294	-.0049	0.0287	0.0717	-.0027	-.0014
-.0181	0.0032	0.0208	-.0461	0.0000	-.0013
0.0091	0.0003	-.0406	0.0188	0.0003	0.0006
-.0038	-.0023	0.0403	0.0026	-.0002	0.0001
0.0019	0.0027	-.0340	-.0138	-.0009	-.0006
-.0018	-.0014	0.0240	0.0177	0.0005	0.0000
0.0006	0.0003	-.0158	-.0190	0.0003	0.0004
0.0005	0.0000	0.0114	0.0184	0.0001	-.0011
-.0008	-.0001	-.0088	-.0176	0.0004	0.0001
0.0000	-.0002	0.0075	0.0165	-.0005	-.0001
0.0001	0.0006	-.0052	-.0148	0.0006	0.0004
0.0002	-.0005	0.0044	0.0126	0.0001	0.0002
0.0000	0.0002	-.0045	-.0110	-.0005	-.0003
-.0002	0.0000	0.0047	0.0097	0.0005	0.0003
0.0003	-.0002	-.0042	-.0017	-.0001	0.0002
0.0002	0.0000	0.0045	0.0062	-.0001	-.0003
-.0004	0.0001	-.0051	-.0040	-.0002	0.0000
0.0003	0.0000	0.0056	0.0023	0.0003	0.0002
0.0002	-.0001	-.0051	-.0008	0.0001	0.0001
-.0002	0.0002	0.0040	-.0007	-.0002	-.0004
0.0000	0.0000	-.0032	0.0019	0.0003	0.0002
0.0001	-.0001	0.0026	-.0026	0.0000	0.0001
0.0001	0.0000	-.0014	0.0025	0.0002	-.0001
-.0003	0.0000	0.0000	-.0023	-.0001	0.0002
0.0002	0.0001	0.0012	0.0014	0.0000	0.0001
0.0000	-.0001	-.0015	-.0007	0.0003	0.0001
0.0000	0.0000	0.0013	-.0002	-.0002	0.0000
0.0000	0.0000	-.0010	0.0008	0.0000	-.0001
-.0001	0.0001	0.0005	-.0009	0.0000	0.0001
0.0002	-.0002	0.0002	0.0007	0.0000	0.0001
0.0000	0.0001	-.0005	-.0002	-.0001	-.0001
-.0002	0.0000	0.0005	-.0002	0.0001	-.0001
0.0001	-.0001	-.0002	0.0005	0.0001	0.0001
0.0000	0.0000	0.0000	-.0005	-.0001	0.0000

Table 2.5 Sampled impulse responses of channels D, E, and F. (chapter 5)

Channel A		Channel B		Channel C	
Real part	Imaginary part	Real part	Imaginary part	Real part	Imaginary part
0.4211	-.4187	-.1998	-.2576	0.9335	-.1289
0.6547	0.2704	0.4609	-.4590	0.3278	0.0183
-.2270	0.2622	0.2913	0.4834	-.1294	-.0182
-.0062	-.1387	-.3389	-.0154	0.0270	-.0006
0.0292	0.0441	0.1233	-.1104	0.0020	0.0011
-.0090	-.0073	-.0143	0.0587	-.0026	-.0013
0.0026	0.0044	0.0040	-.0153	-.0012	0.0008
-.0053	-.0010	-.0062	0.0117	-.0011	-.0003
0.0057	-.0011	0.0016	-.0078	0.0018	0.0007
-.0031	0.0001	0.0026	0.0040	-.0012	0.0002
0.0006	0.0006	-.0007	0.0007	-.0009	-.0003
-.0007	-.0040	-.0014	0.0002	0.0015	-.0003
0.0007	0.0015	0.0001	0.0000	-.0005	0.0001
-.0002	-.0021	-.0014	0.0022	-.0006	0.0003
-.0005	-.0002	-.0001	-.0026	0.0006	-.0002
0.0013	0.0002	0.0020	-.0002	0.0000	0.0004
-.0012	0.0001	0.0004	0.0022	0.0002	-.0004
0.0002	0.0001	-.0014	-.0012	-.0004	0.0000
0.0000	0.0001	0.0012	-.0004	0.0000	0.0002
0.0002	-.0001	0.0003	0.0001	0.0004	-.0002
0.0001	0.0008	0.0002	0.0003	-.0001	0.0002
-.0005	0.0003	-.0007	0.0000	-.0002	-.0001
0.0001	-.0001	-.0002	-.0008	0.0002	0.0001
0.0004	0.0005	0.0006	-.0003	0.0003	-.0001
-.0002	0.0001	0.0000	0.0004	-.0004	-.0001
0.0001	-.0001	-.0002	-.0001	0.0000	0.0002
0.0000	-.0004	0.0001	-.0001	0.0002	-.0001
0.0002	-.0001	0.0000	0.0000	0.0000	0.0001
-.0003	0.0000	0.0000	0.0000	-.0002	0.0000
0.0000	-.0004	0.0000	0.0000	0.0001	0.0000
0.0003	-.0001			0.0001	0.0000
-.0001	0.0001			-.0002	-.0001
-.0002	-.0001			0.0001	0.0001
-.0001	0.0000			0.0000	-.0001
0.0001	-.0001			0.0001	0.0000
0.0002	0.0002			0.0000	0.0000
-.0001	0.0003			-.0002	0.0000
0.0000	0.0000			0.0002	0.0000
0.0001	0.0001			0.0001	0.0000
0.0000	0.0001			-.0001	0.0000
0.0000	-.0001			0.0001	0.0000
0.0001	0.0000			0.0000	0.0000
0.0000	0.0001			-.0001	0.0000
0.0000	0.0000			-.0001	0.0000
0.0000	0.0000			0.0000	0.0000

Table 2.6 Sampled impulse responses obtained by replacing all zeros of the corresponding z-transfrom of channels A - C (Table 2.4) that lie outside the unit circle in the z-plane, by the complex conjugates of their reciprocals.

Channel D		Channel E		Channel F	
Real part	Imaginary part	Real part	Imaginary part	Real part	Imaginary part
0.8702	-.0360	-.8367	0.2919	-.5201	-.6667
0.4450	0.1566	-.3848	0.1897	-.0339	-.5105
-.1262	-.0020	0.1682	-.0160	0.1060	0.0972
0.0257	-.0095	-.0410	0.0048	-.0352	0.0086
0.0084	0.0064	0.0024	-.0030	0.0185	-.0078
-.0120	-.0029	-.0074	0.0117	-.0012	0.0166
0.0089	0.0002	0.0138	-.0103	-.0034	-.0059
-.0095	-.0008	-.0059	0.0100	0.0023	0.0034
0.0073	0.0018	0.0058	-.0104	-.0034	-.0041
-.0039	-.0016	-.0060	0.0081	0.0034	0.0007
0.0002	0.0021	0.0086	-.0077	-.0007	0.0019
0.0011	-.0019	-.0082	0.0079	-.0008	-.0006
-.0003	0.0017	0.0065	-.0072	0.0002	0.0002
-.0007	-.0007	-.0052	0.0072	0.0008	0.0007
0.0001	-.0002	0.0042	-.0069	-.0008	0.0001
0.0002	0.0005	-.0040	0.0066	0.0001	-.0008
-.0001	-.0004	0.0021	-.0060	-.0007	0.0005
-.0003	-.0001	-.0003	0.0060	-.0004	0.0000
-.0003	0.0003	-.0006	-.0058	0.0005	-.0003
0.0004	-.0001	0.0010	0.0049	-.0004	-.0002
0.0001	0.0001	-.0025	-.0039	0.0002	-.0004
-.0003	-.0001	0.0032	0.0022	0.0004	0.0004
0.0002	0.0000	-.0034	-.0012	-.0003	-.0003
0.0003	0.0001	0.0026	0.0000	0.0001	-.0002
-.0003	-.0002	-.0019	0.0012	0.0001	0.0004
0.0000	0.0001	0.0015	-.0021	0.0002	0.0000
0.0004	-.0001	-.0009	0.0023	-.0003	-.0002
-.0001	0.0002	-.0002	-.0018	0.0001	-.0002
-.0001	-.0001	0.0012	0.0013	0.0001	0.0030
0.0001	0.0001	-.0013	-.0003	-.0002	-.0002
0.0002	0.0000	0.0009	-.0002	0.0000	-.0002
-.0002	-.0001	-.0004	0.0008	-.0001	0.0001
0.0001	0.0001	0.0001	-.0010	0.0002	-.0002
0.0000	0.0000	0.0003	0.0007	0.0001	0.0000
0.0001	-.0001	-.0006	-.0002	-.0001	-.0002
0.0000	0.0001	0.0004	-.0002	0.0001	0.0000
-.0002	0.0000	-.0001	0.0003	0.0000	0.0001
0.0002	0.0000	-.0002	-.0003	0.0000	-.0001
0.0000	0.0000	0.0001	0.0001	0.0000	0.0000
-.0001	0.0000	0.0001	0.0002	0.0001	0.0001
0.0000	0.0000	0.0000	0.0000	-.0002	0.0001
0.0000	0.0000	0.0000	0.0000	-.0001	-.0001
0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 2.7 Sampled impulse responses obtained by replacing all zeros of the corresponding z-transform of channels D - F (Table 2.5) that lie outside the unit circle in the z-plane, by the complex conjugates of their reciprocals.

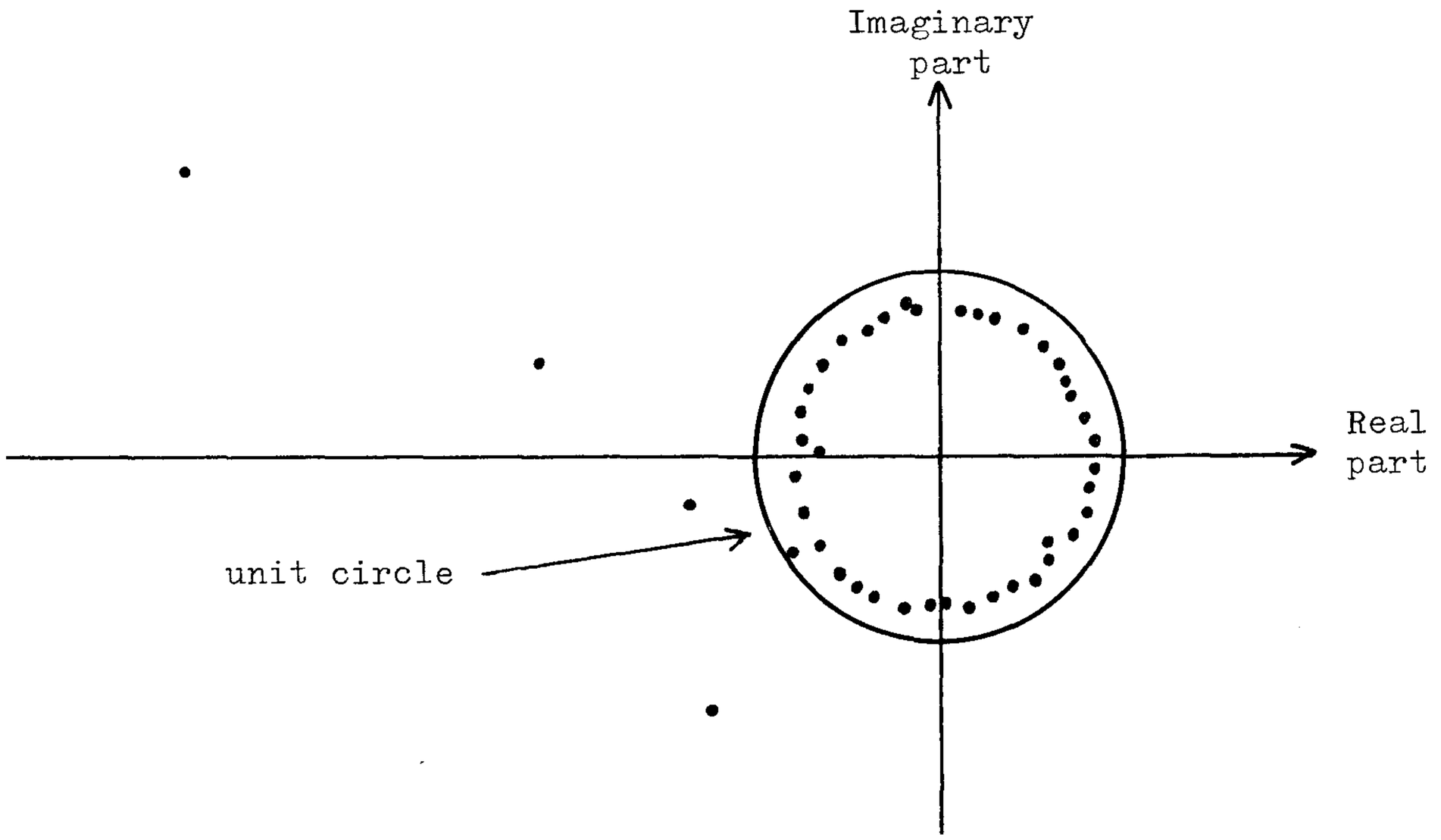


Fig. 2.5 Zeros (or roots) of the z-transform of channel A (Table 2.4).

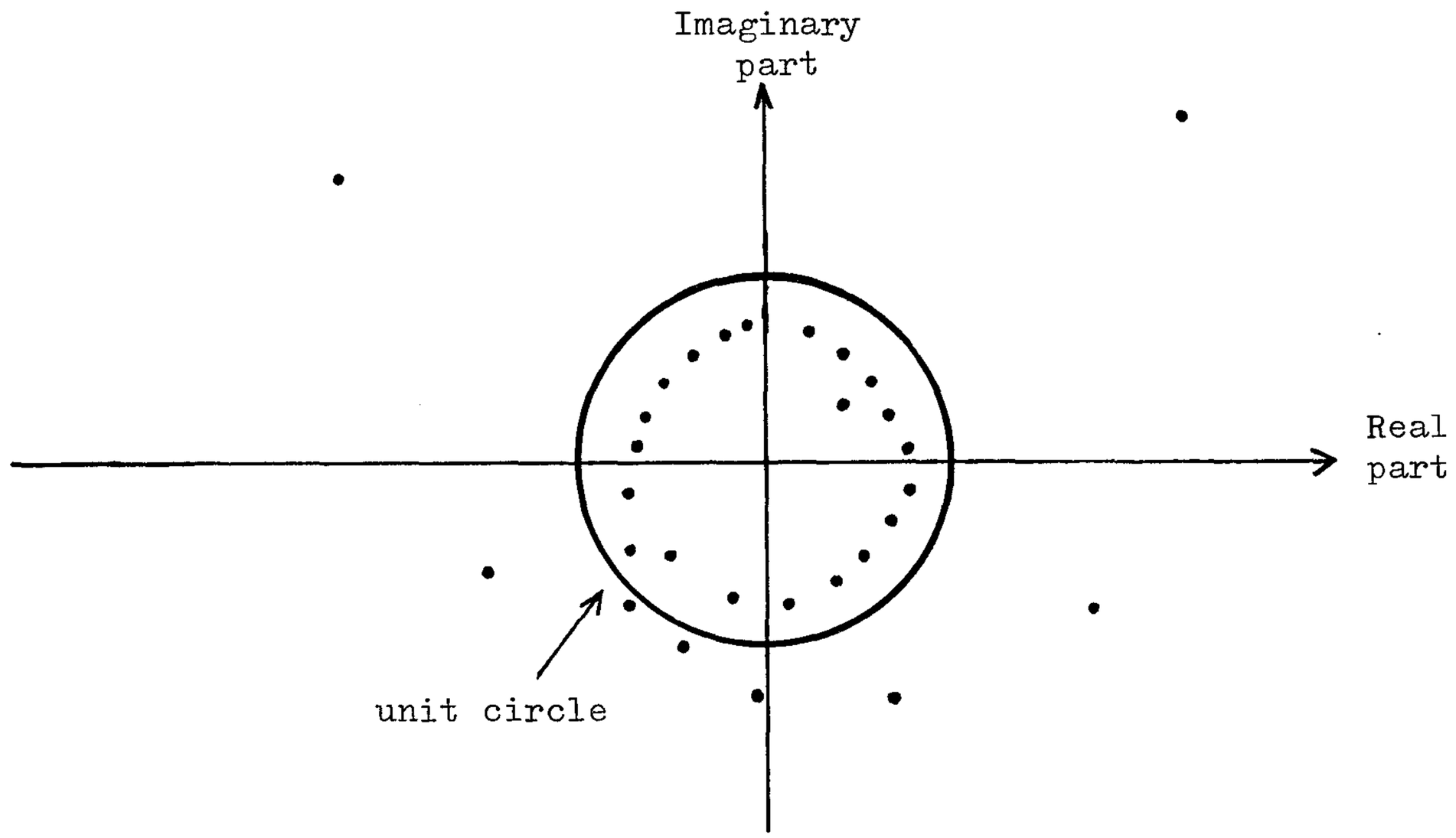


Fig. 2.6 Zeros (or roots) of the z-transform of channel B (Table 2.4).

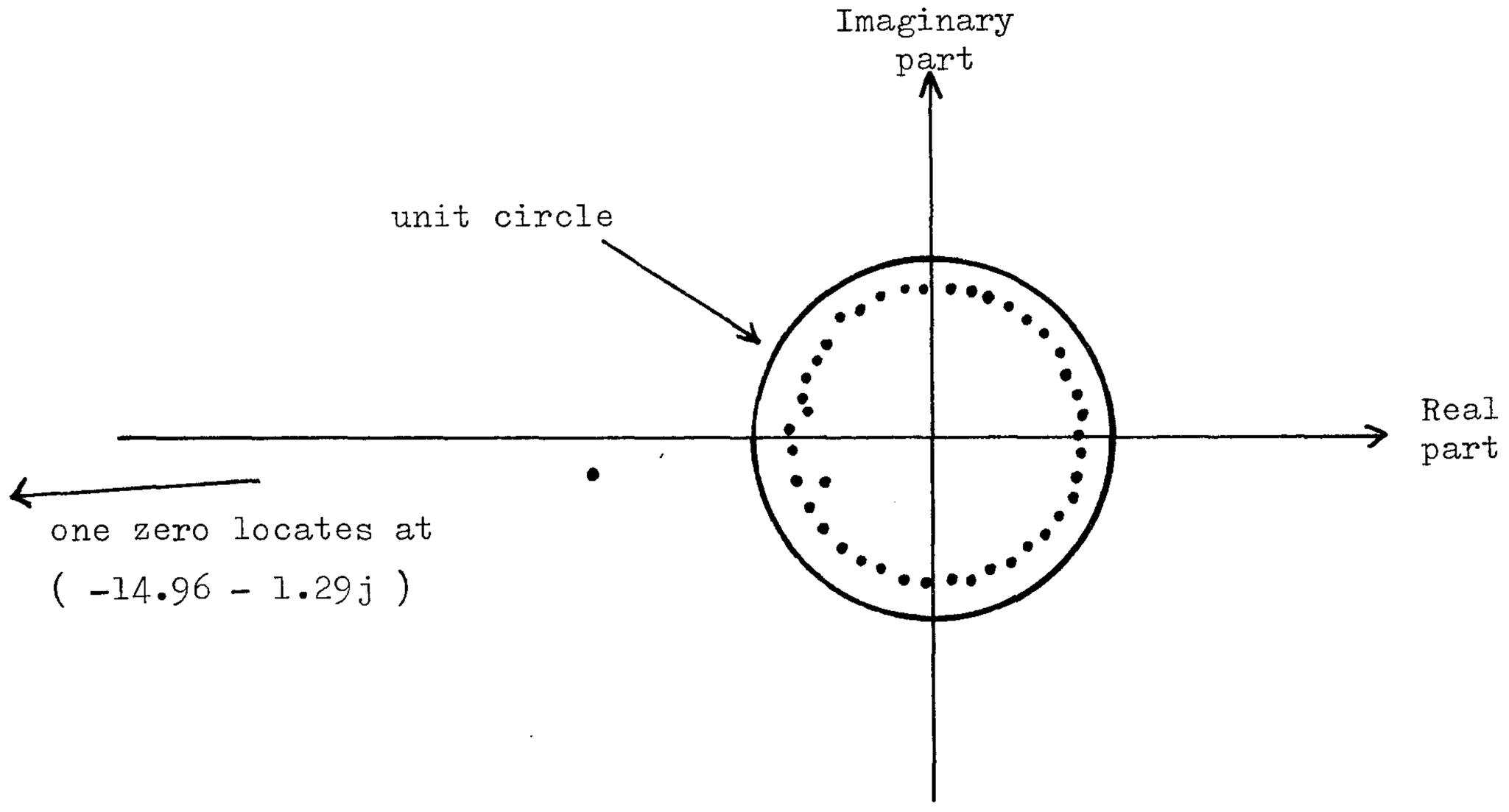


Fig. 2.7 Zeros (or roots) of the z-transform of channel C (Table 2.4).

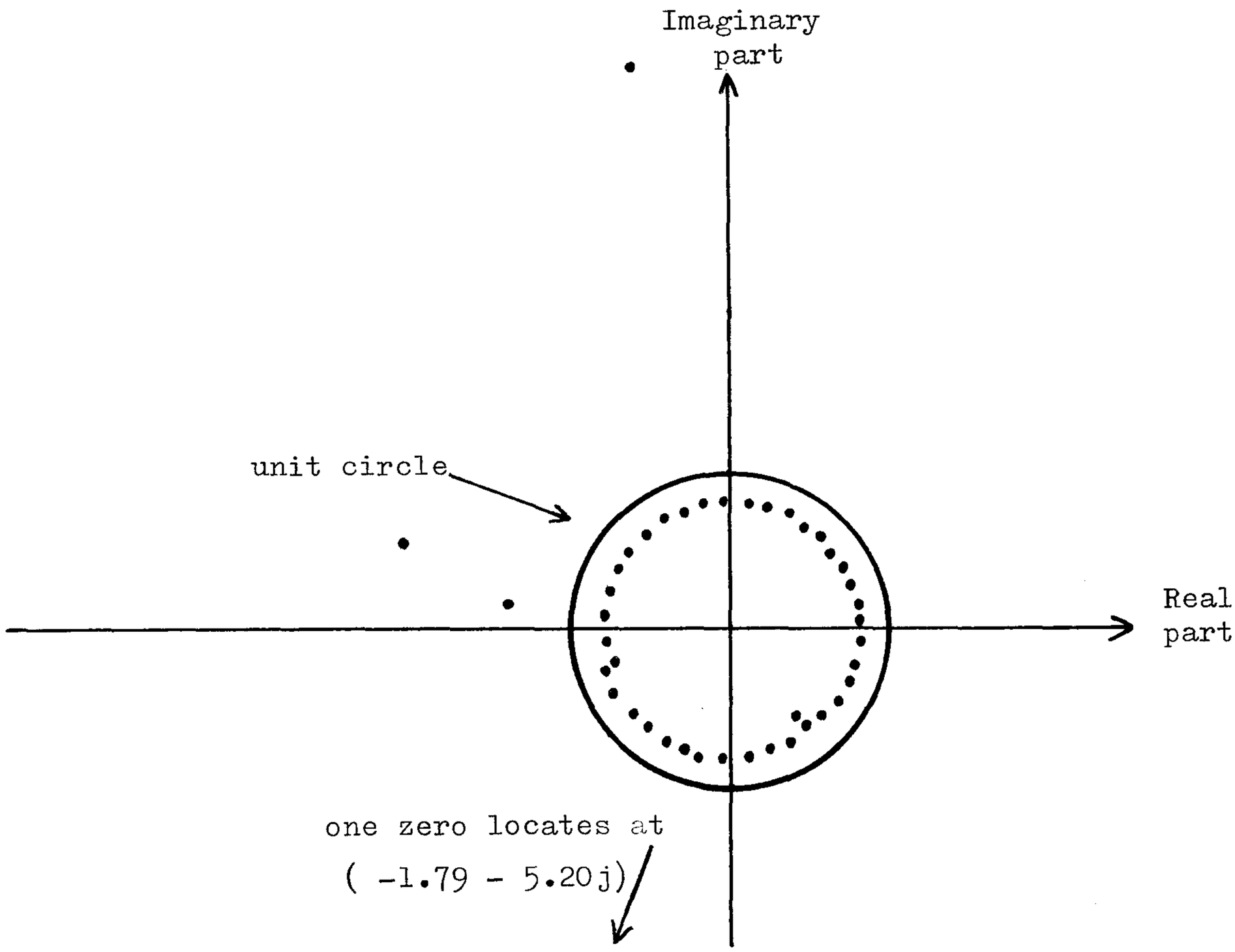


Fig. 2.8 Zeros (or roots) of the z-transform of channel D (Table 2.5).

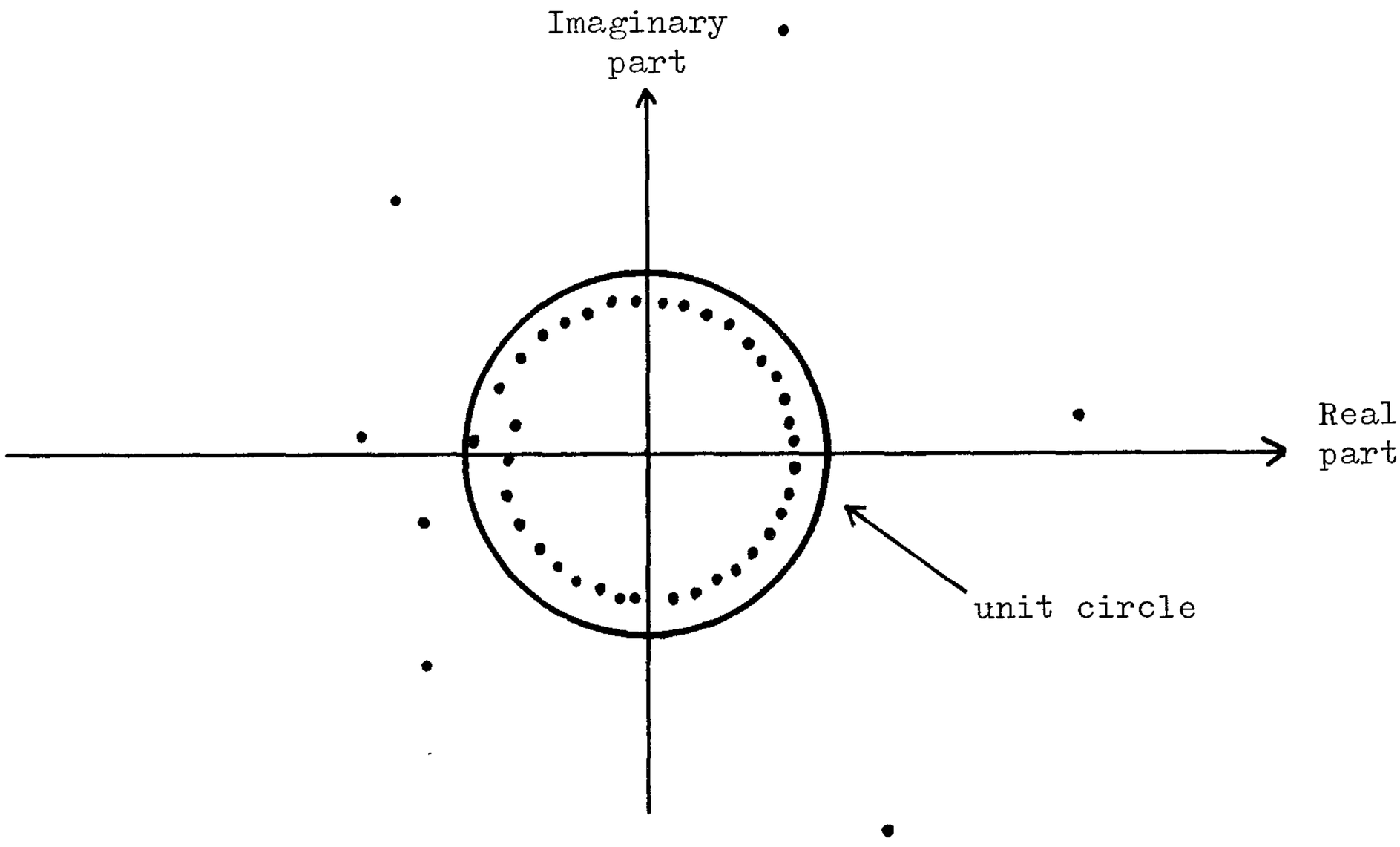


Fig. 2.9 Zeros (or roots) of the z-transform of channel E (Table 2.5).

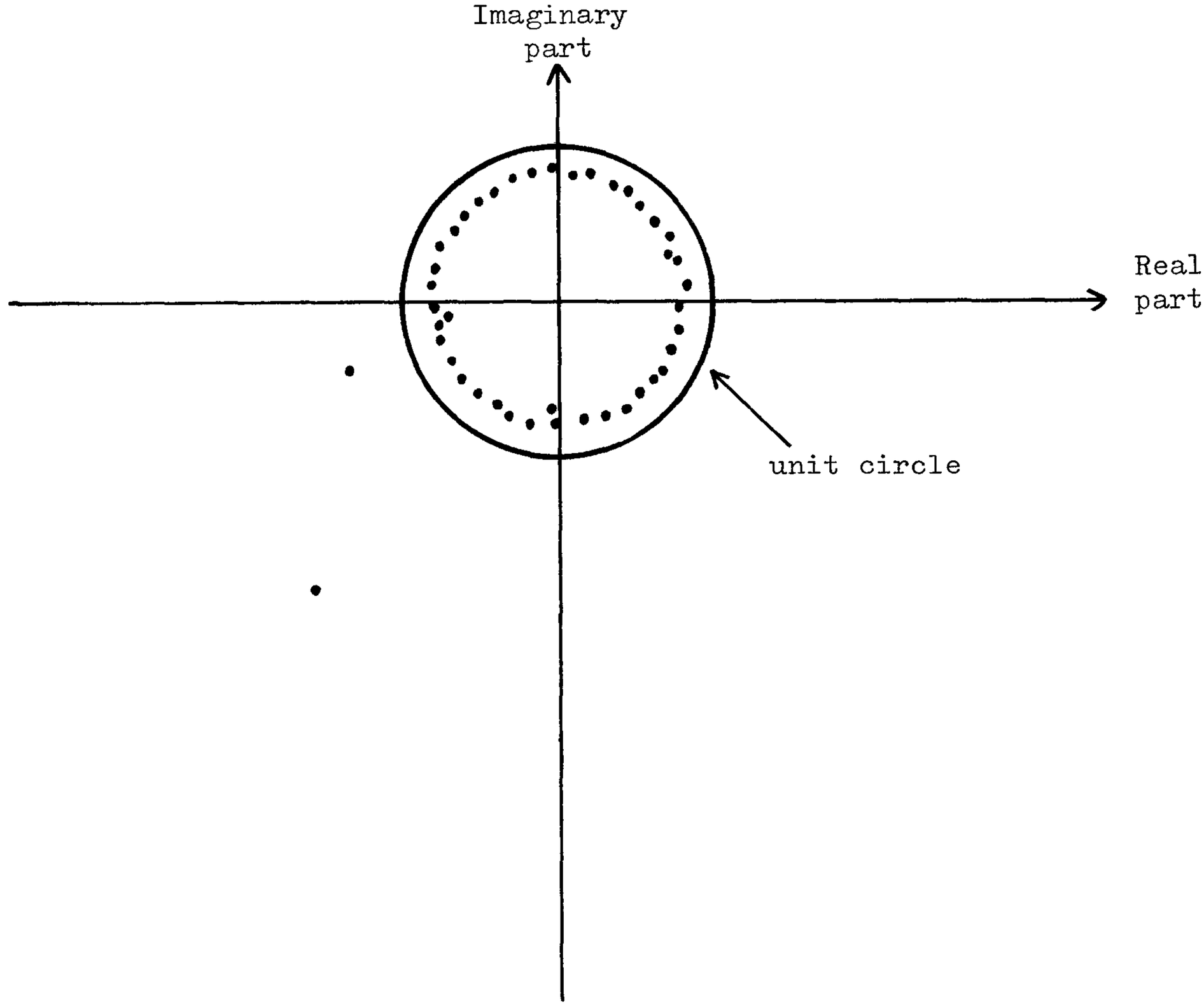
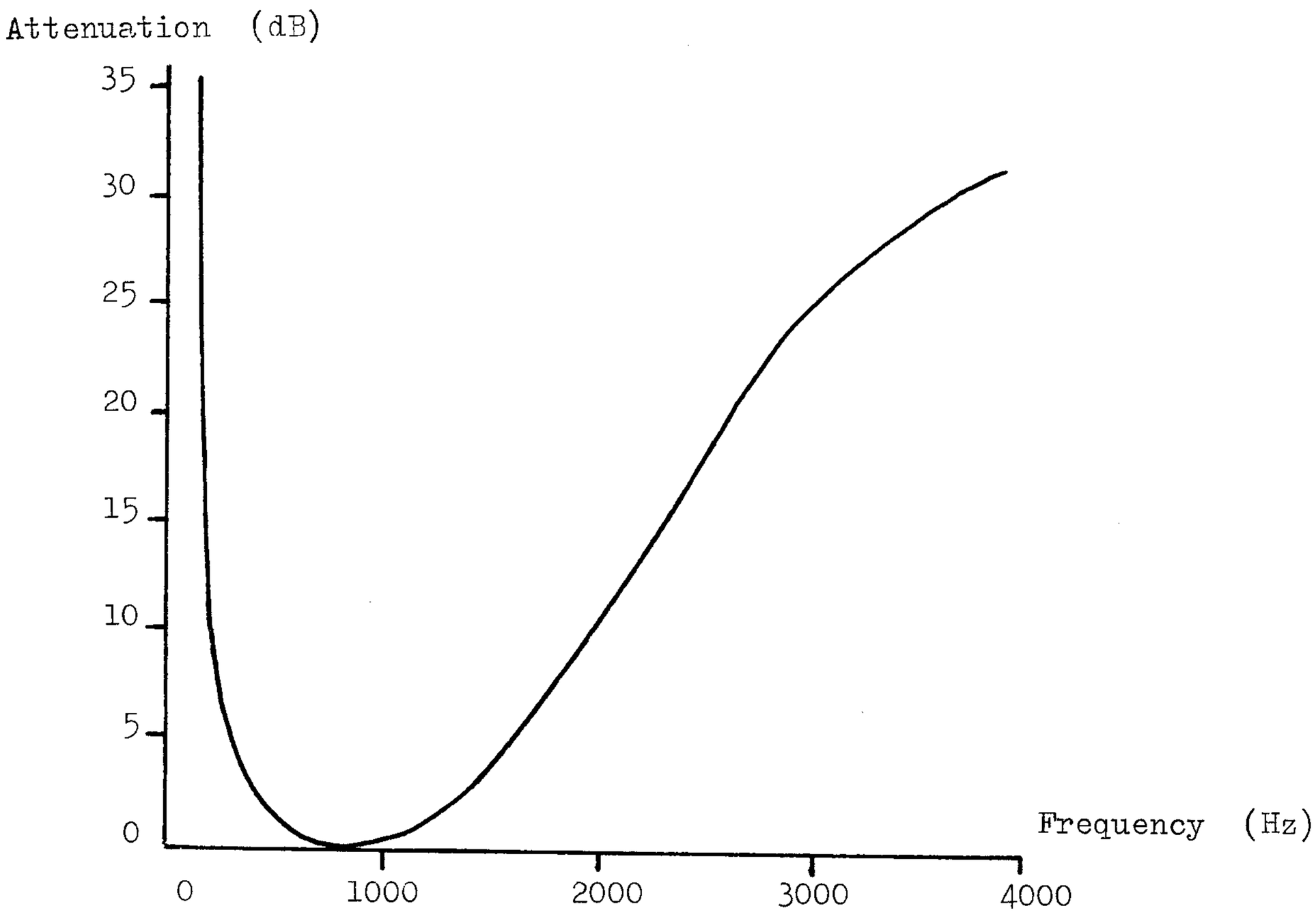


Fig. 2.10 Zeros (or roots) of the z-transform of channel F (Table 2.5).



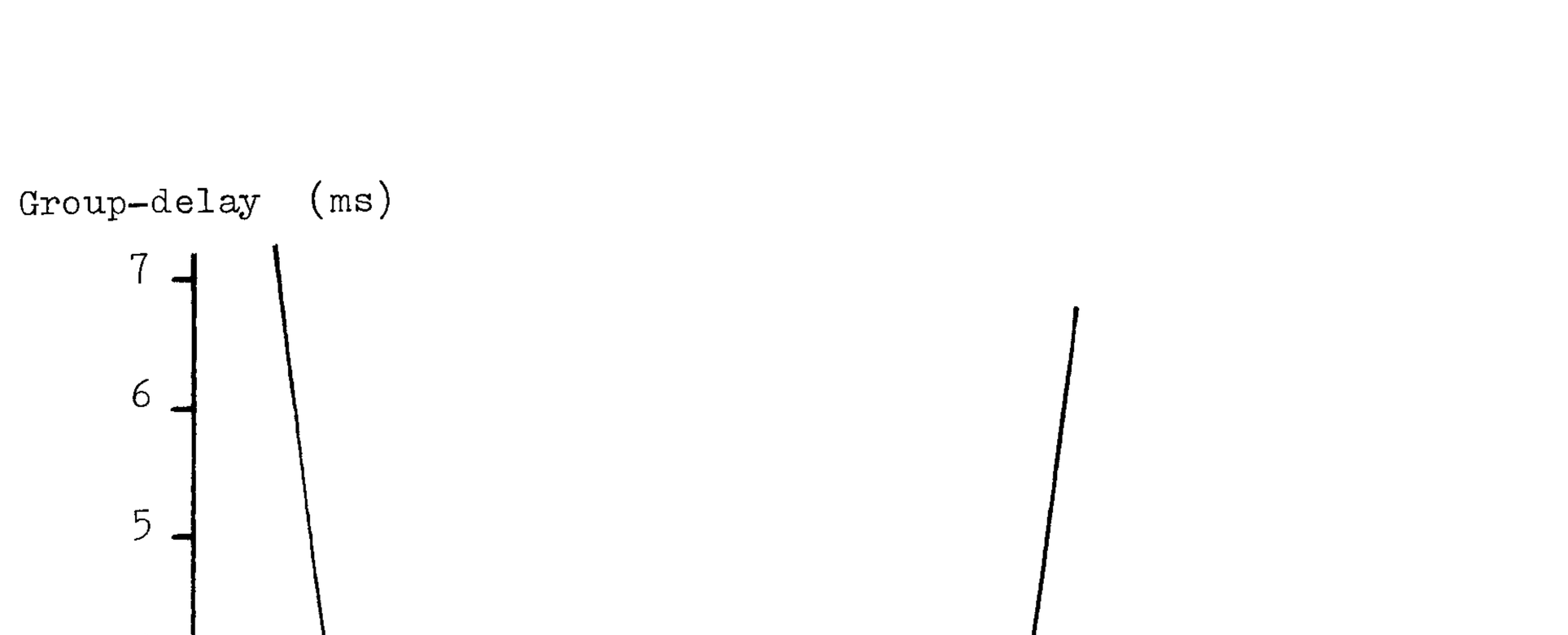
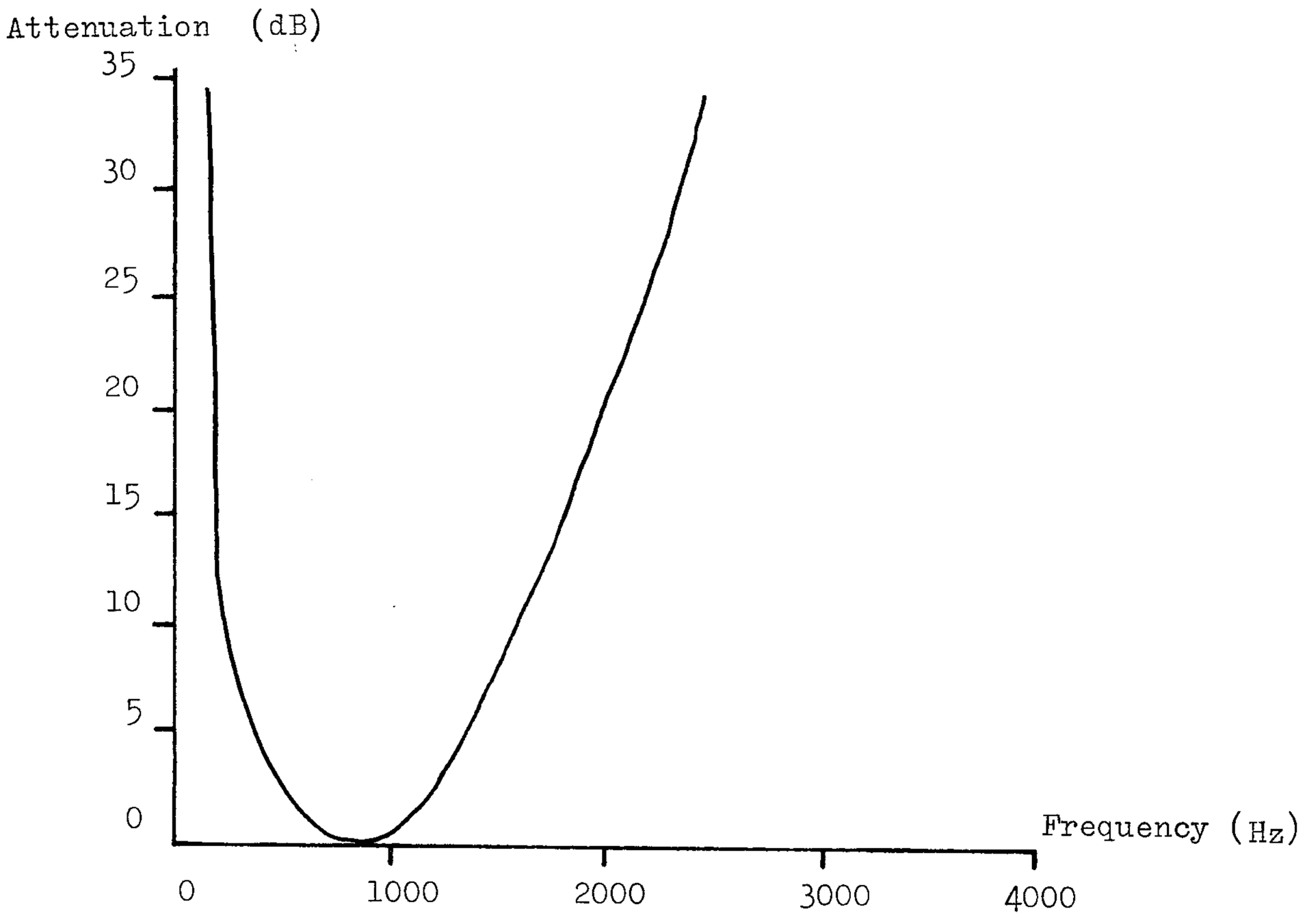


Fig. 2.11 Attenuation and Group-delay charateristics of telephone circuit A (Table 2.4).



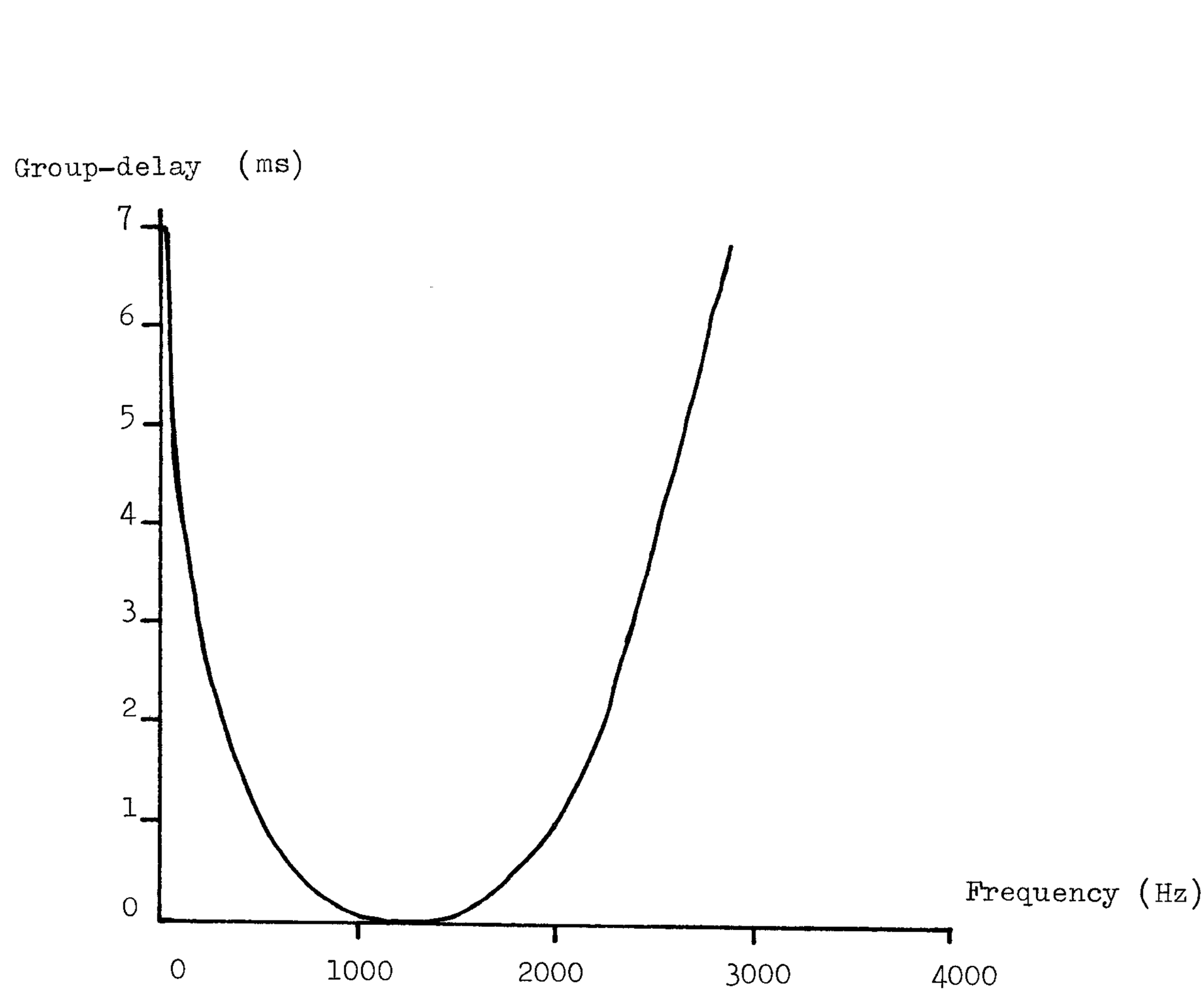


Fig. 2.12 Attenuation and Group-delay charateristics of telephone circuit B (Table 2.4).

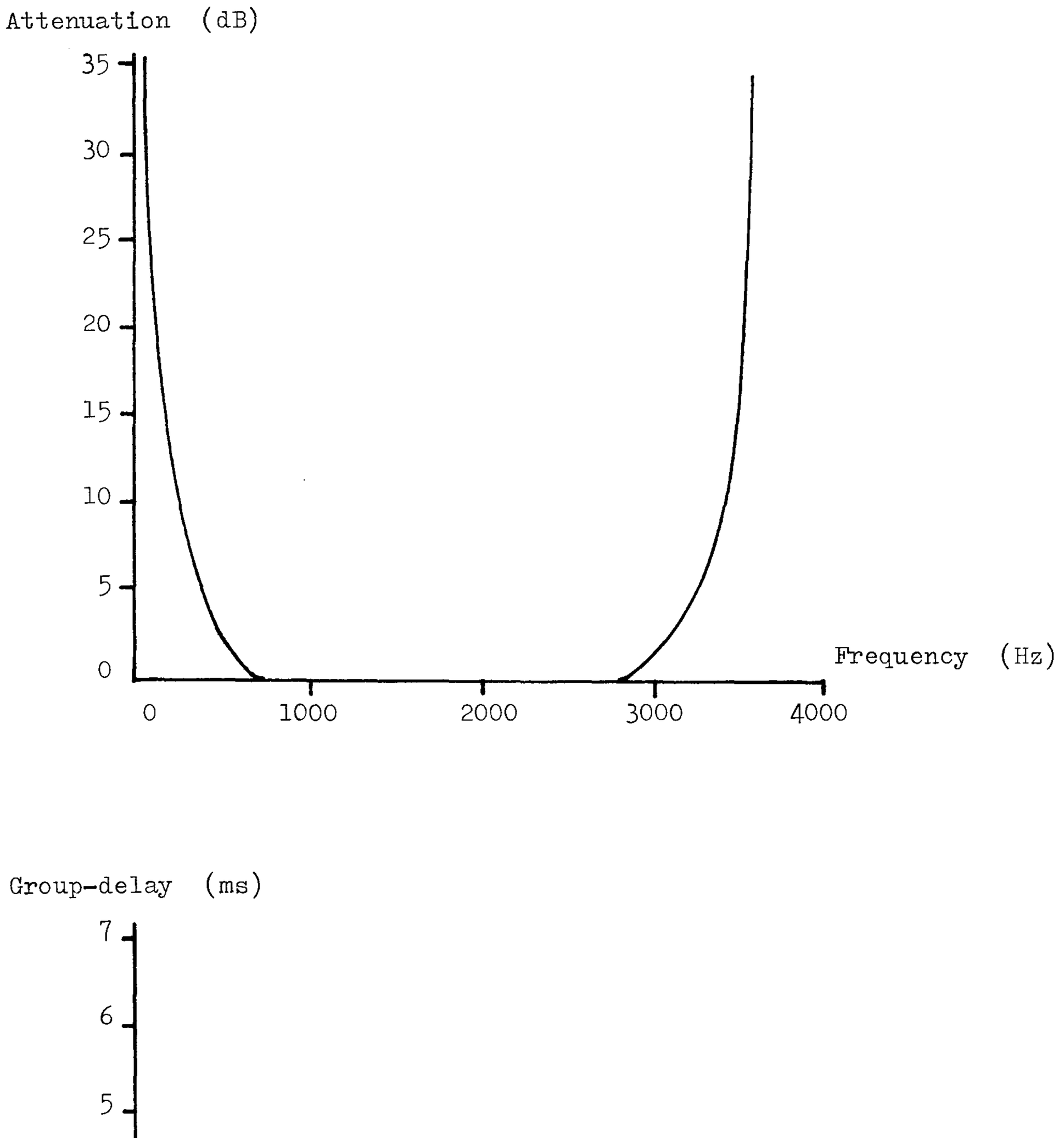


Fig. 2.13 Attenuation and Group-delay charateristics of telephone circuit C (Table 2.4).

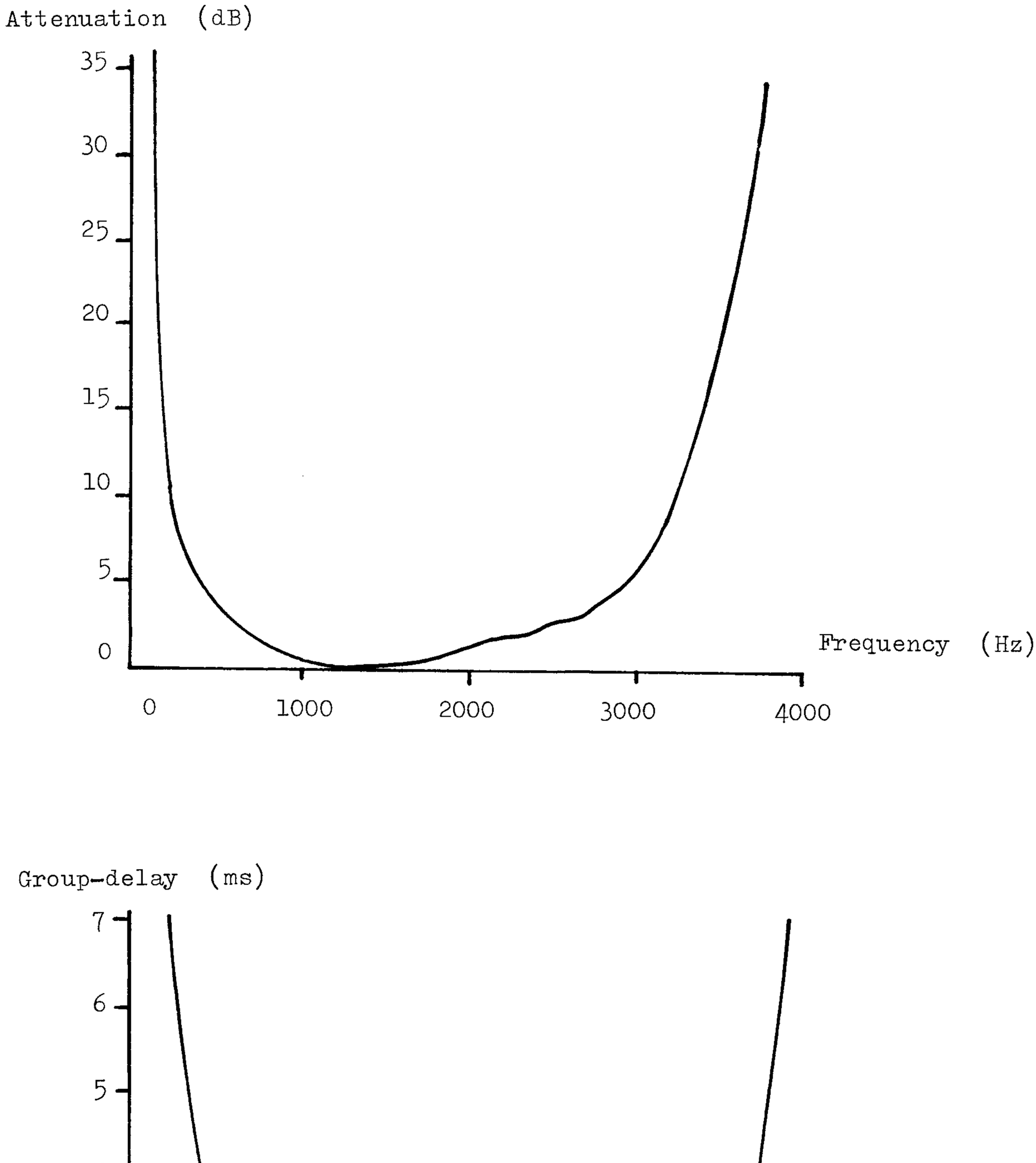


Fig. 2.14 Attenuation and Group-delay charateristics of telephone circuit D (Table 2.5).

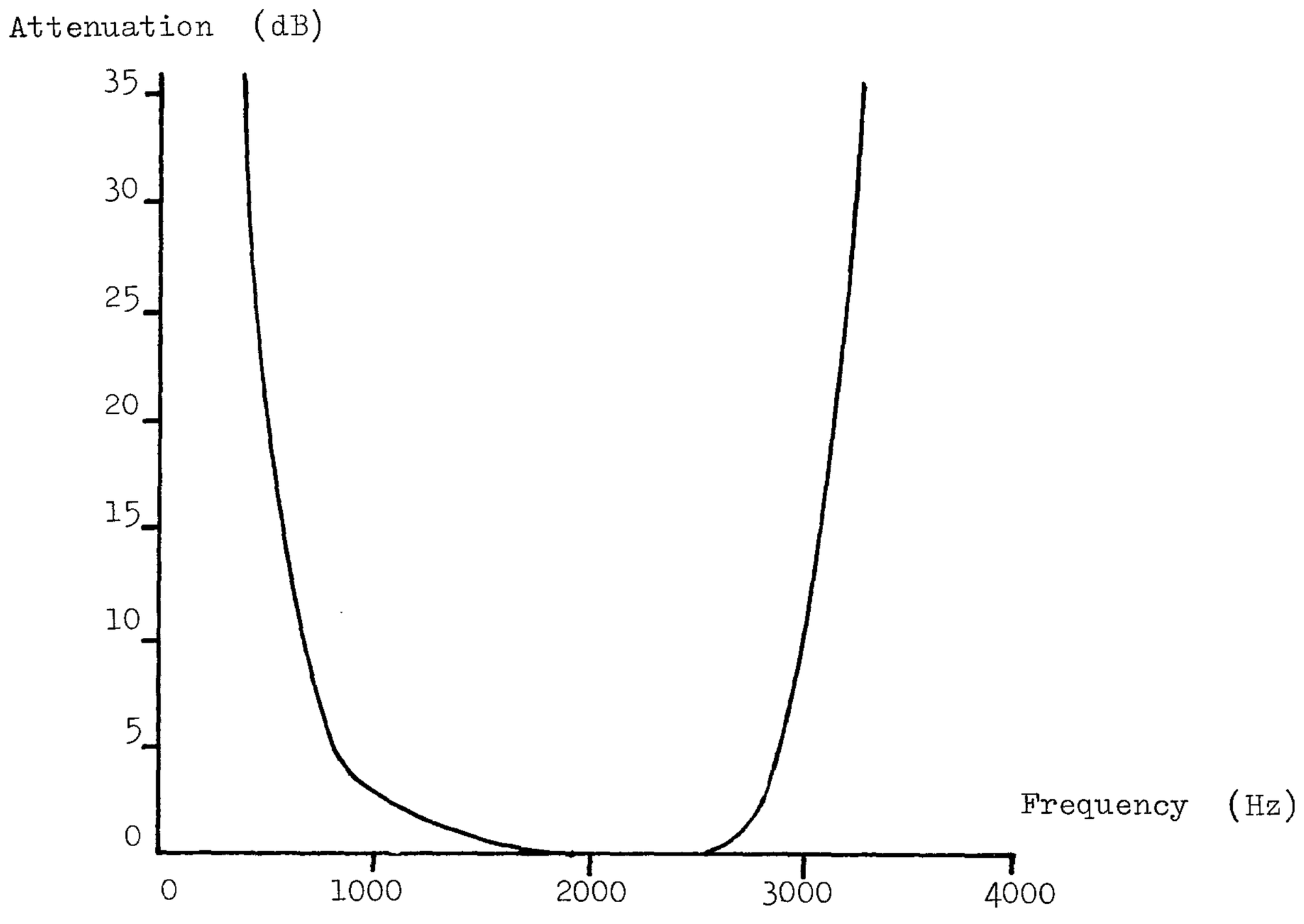


Fig. 2.15 Attenuation and Group-delay charateristics of telephone circuit E (Table 2.5).

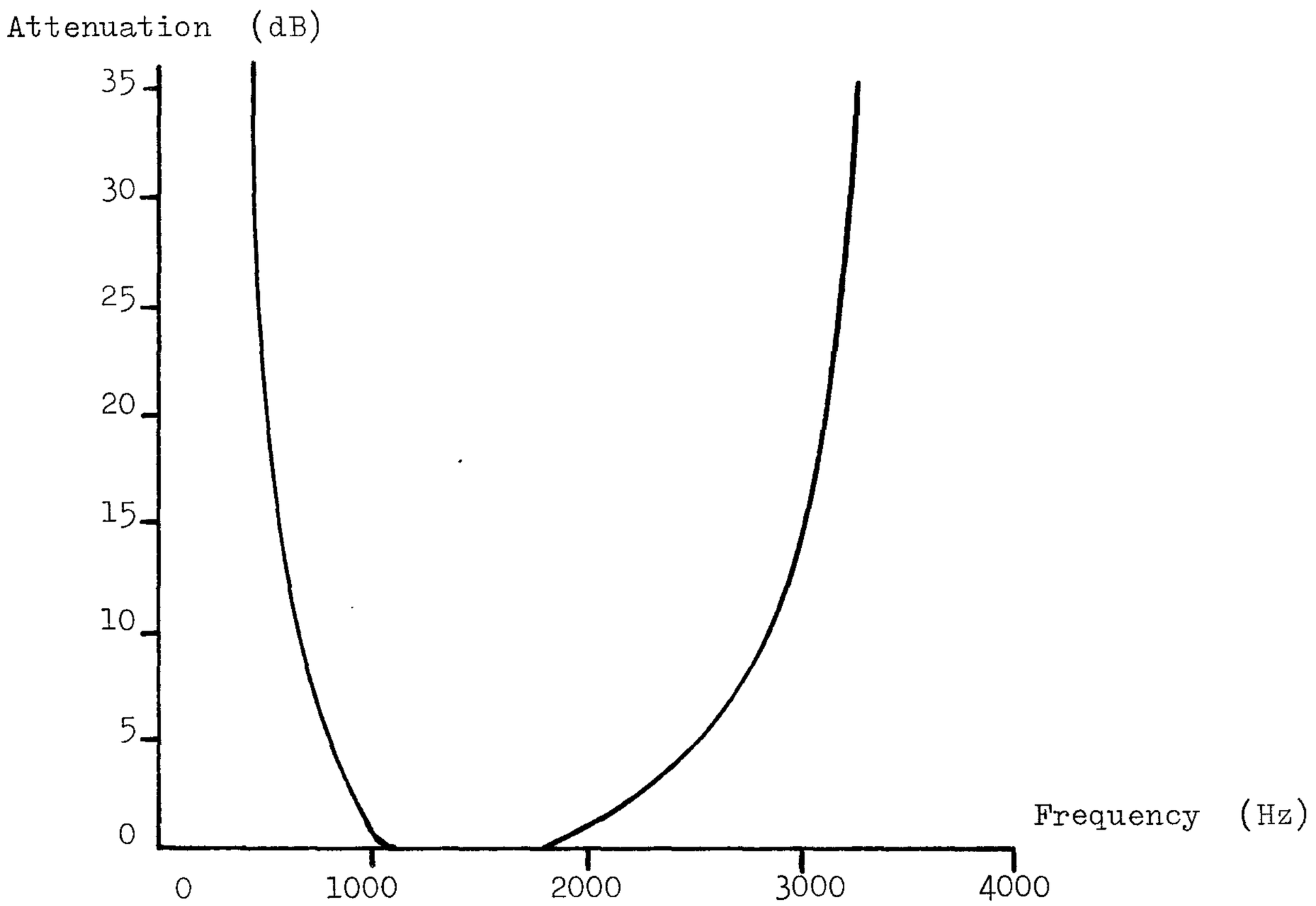
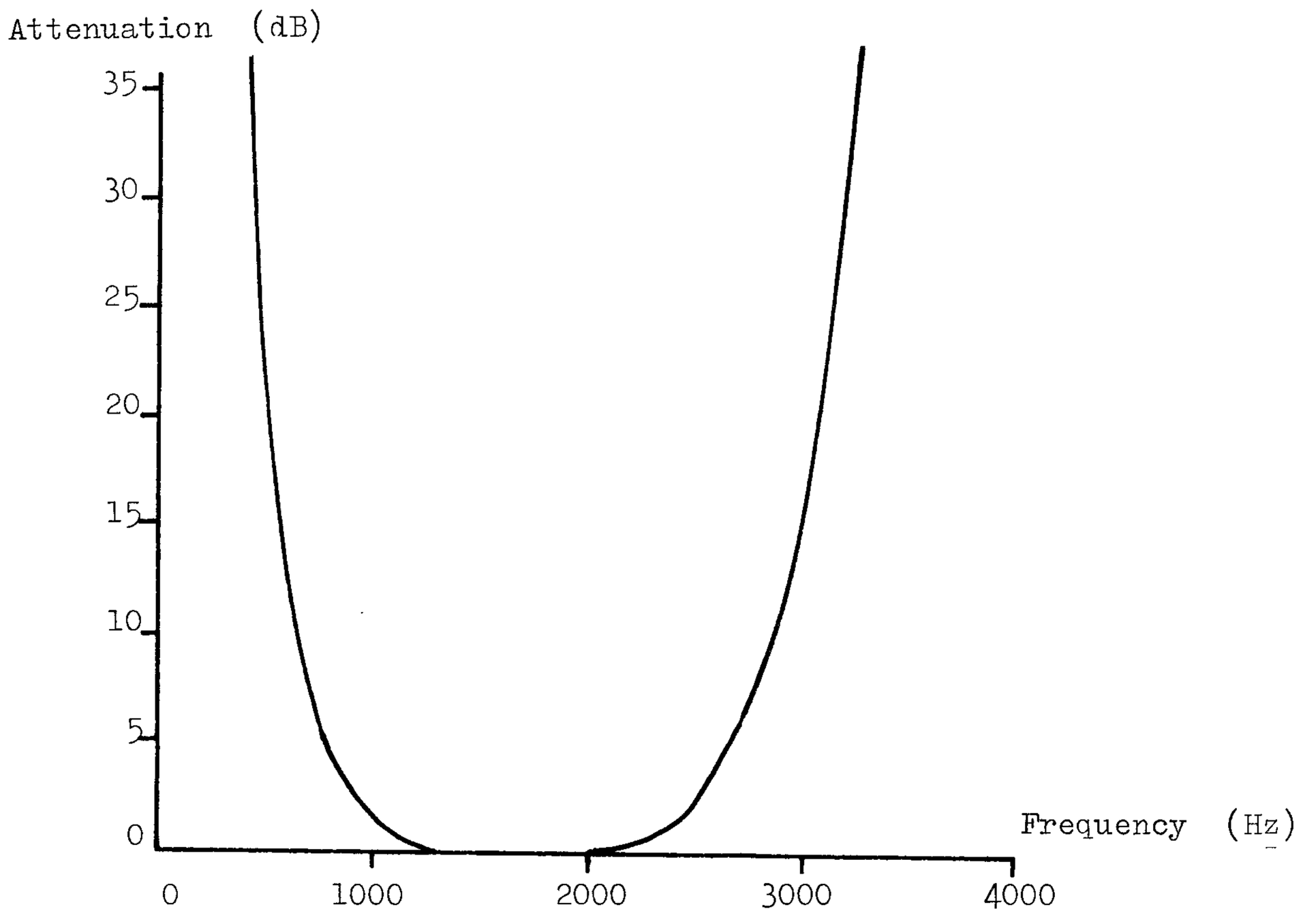


Fig. 2.16 Attenuation and Group-delay charateristics of telephone circuit F (Table 2.5).



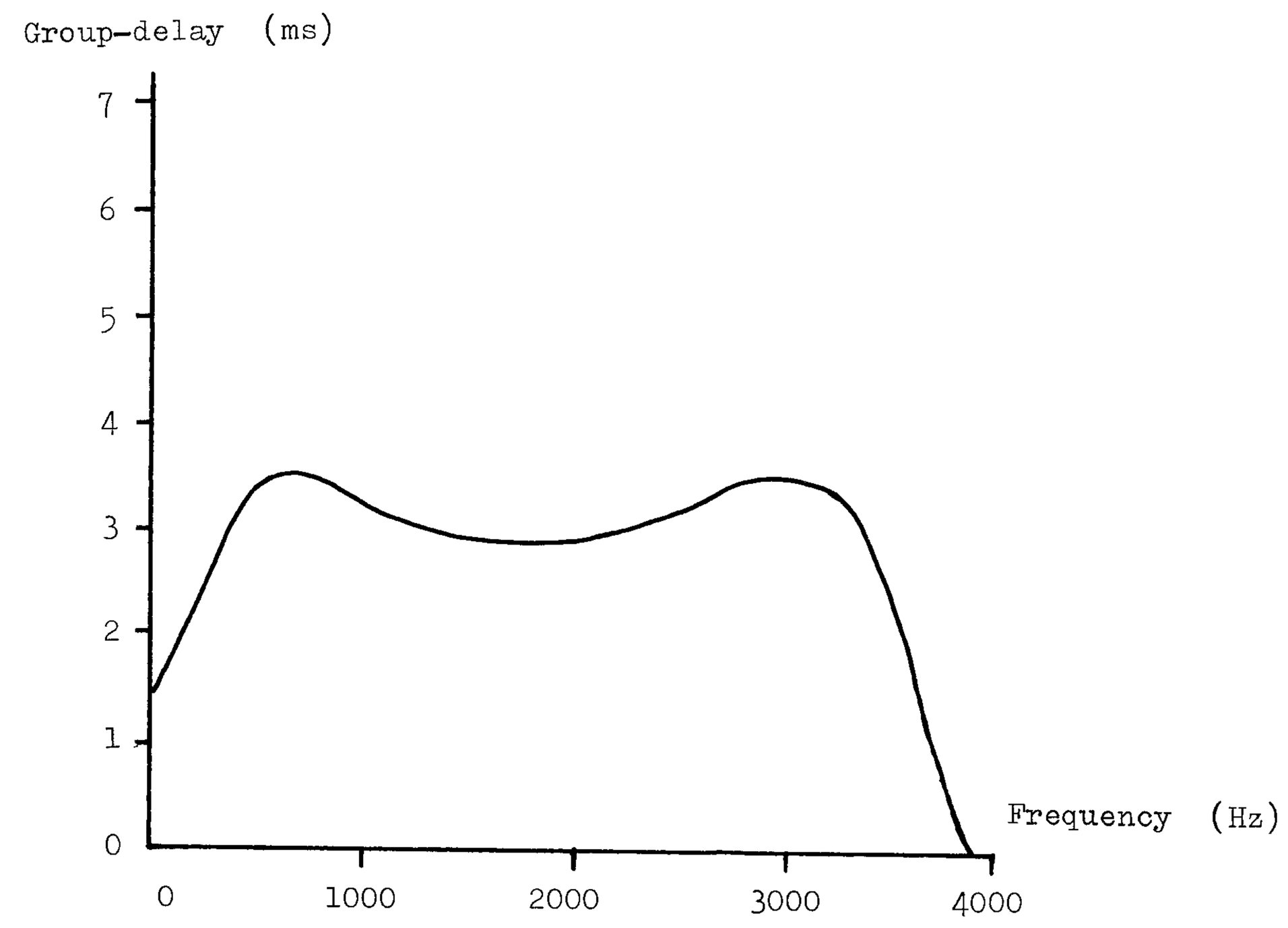


Fig. 2.17 Attenuation and Group-delay charateristics of the combination of transmitter and receiver filters (Fig. 2.3).

2.6 Channel Equalization

In the data-transmission system shown in Fig. 2.1 or 2.3, the baseband channel is time-dispersive in that the channel impulse response has, for practical purposes, a duration of $(g+1)T$ seconds where g is a positive integer and T seconds is the sampling interval. Thus, if the sampled impulse response of the channel is the sequence of values $y_0, y_1, \cdots, y_g$ then the components of an individual received signal-element at the output of the sampler in Fig. 2.1 or 2.3 are $s_i y_0, s_i y_1, \cdots, s_i y_g$ which are received, in turn, at $(g+1)$ consecutive sampling instants. s_i is here the data symbol carried by the ith transmitted signal element. Consequently, the received sample r_i at the output of the sampler contains not only the data symbol s_i, but also the previous data symbols $s_{i-1}, s_{i-2}, \cdots, s_{i-g}$. This is the effect of intersymbol interference.

Various equalizers have been designed to eliminate the intersymbol interference introduced by the channel;[B1-B37] but only the more fundamental and important types of equalizers are described here. Furthermore, the equalizers to be described here are for use with the QAM signals where the signals are represented by complex-valued quantities. The corresponding equalizers for use with the binary signals can easily be derived from here by setting to zero all imaginary parts of the complex signals.

2.6.1 Linear equalizer

The signal processor in Fig. 2.3 is here implemented as a linear equalizer followed by a threshold detector shown in Fig. 2.18. The most widely studied linear equalizer is the linear feedforward transversal equalizer (filter) as shown in Fig. 2.19.

The signals marked in Fig. 2.19 are those present at the time instant $t = iT$. Each block marked with the symbol T is a delay unit introducing a time delay of T seconds which is the sampling interval. For an equalizer having p+1 taps, there are altogether p+1 stores holding the received samples r_{i-p}, $\cdots$, r_{i-1}, r_i which are here complex-valued quantities. A multiplier 'cell' is associated with each of these p+1 stores and it multiplies the complex-valued quantity r_{i-h} by a complex-valued quantity f_h, so that the resultant signal at the output of the equalizer is the complex-valued quantity

$$r'_i = \sum_{h=0}^{p} r_{i-h} \, f_h \qquad (2.21)$$

where $f_0, f_1, \cdots, f_p$ are the tap gains of this linear equalizer.

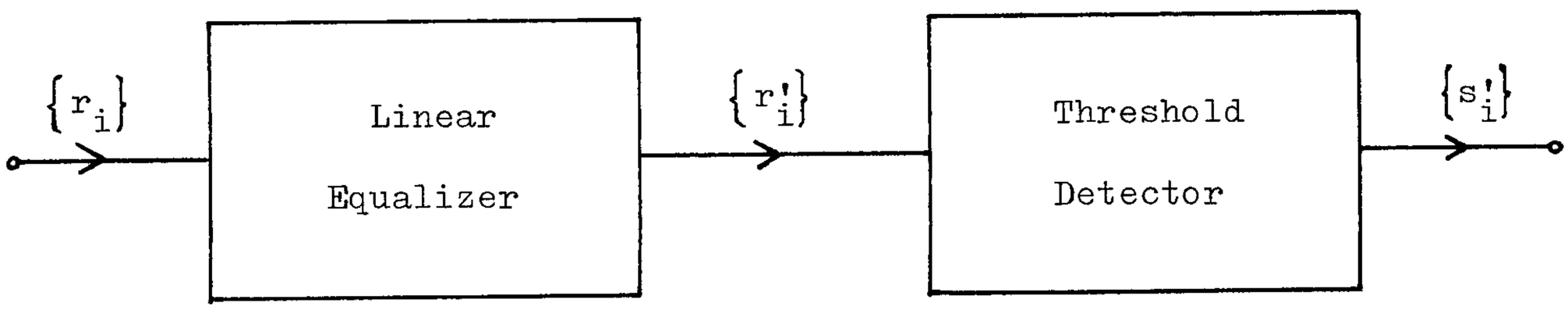


Fig. 2.18 Signal processor implemented as a combination of a linear equalizer and a threshold detector.

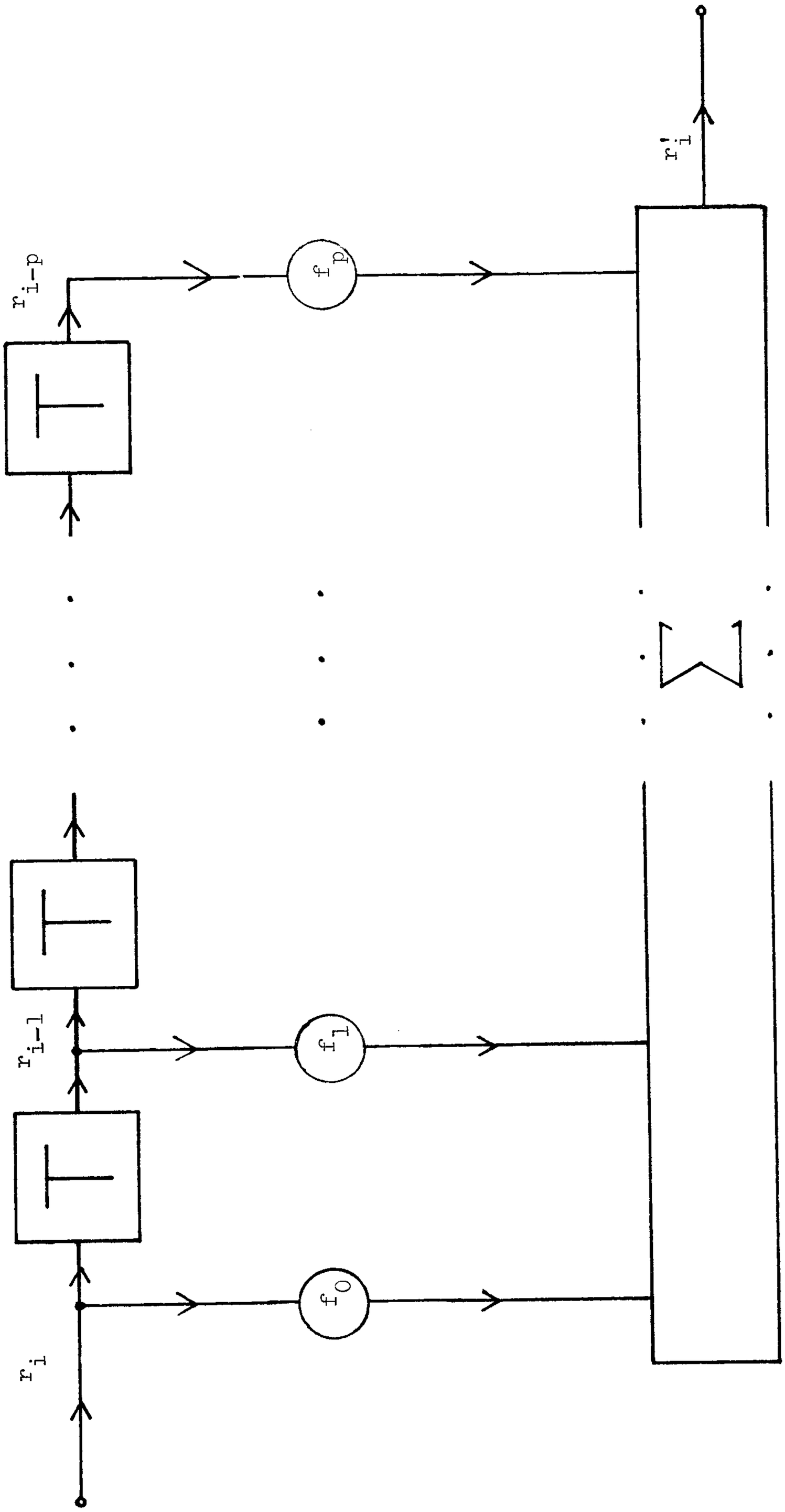


Fig. 2.19 A p-tap linear feedforward transversal equalizer.

Let V(z) be the z-transform of the channel (Fig. 2.3) and let F(z) be the z-transform of the linear equalizer (Fig. 2.19). Thus,

$$V(z) = y_0 + y_1 z^{-1} + \cdots + y_g z^{-g} \tag{2.22}$$

and

$$F(z) = f_0 + f_1 z^{-1} + \cdots + f_p z^{-p} \tag{2.23}$$

where the $\{y_i\}$ and $\{f_i\}$ are as defined in eqns. 2.18 and 2.21, respectively. The z-transform of the channel and linear equalizer is therefore

$$V'(z) = V(z)F(z) \tag{2.24}$$

and so long as the channel is accurately equalized, then[A9]

$$V'(z) \simeq (1+j0)\, z^{-k} = z^{-k} \tag{2.25}$$

where $j = \sqrt{-1}$ and k is a positive integer between 0 and p+g.

Eqn. 2.25 implies that the baseband channel, the sampler, and the linear equalizer, together form a network that introduces only a delay of kT seconds onto the transmitted data symbols $\{s_i\}$. The sample value at the output of the equalizer, at time t = (i+k)T, is thus

$$r'_{i+k} \simeq s_i + w'_{i+k} \tag{2.26}$$

where

$$w'_{i+k} = \sum_{h=0}^{p} w_{i+k-h} f_h \tag{2.27}$$

and $\{w_i\}$ are the noise components at the output of the sampler in Fig. 2.3. Since the real and imaginary parts of the $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and variance σ^2, therefore the real and imaginary parts of the noise

components $\{w'_i\}$ (eqn. 2.26) are also Gaussian random variables with zero mean and variance[A9]

$$\eta^2 = \sigma^2 \sum_{h=0}^{p} |f_h|^2 \qquad (2.28)$$

where $|f_h|$ is the modulus of the complex-valued quantity f_h.

The threshold detector (Fig. 2.18) operates on the sample value r'_{i+k} (eqn. 2.26) to give the detected value of the data symbol s_i. This is carried out according to Table 2.8 for a 16-point QAM signal. The threshold values used in Table 2.8 are such that the probability of error in the detection of a_i from $\mathrm{Re}(r'_{i+k})$, or b_i from $\mathrm{Im}(r'_{i+k})$, is minimized,[A9] where a_i and b_i are the real and imaginary parts of s_i respectively, and Re(.) and Im(.) are the real and imaginary parts of (.) respectively. Under the various conditions assumed here, it can be shown that[A8] the average probability of error in the detection of a_i or b_i is given by

$$\left(\frac{3}{2}\right)\int_1^{\infty} \frac{1}{\sqrt{2\pi}\eta} \exp\left(\frac{-u^2}{2\eta^2}\right) du \;=\; \frac{3}{2}\; Q\left(\frac{1}{\eta}\right) \qquad (2.29)$$

where η^2 is as given by eqn. 2.28 and Q(.) is the well-known Q-function.

a_i or b_i = −3	if	$-2 \geqslant \mathrm{Re}(r'_{i+k})$	or $\mathrm{Im}(r'_{i+k})$
a_i or b_i = −1	if	$0 \geqslant \mathrm{Re}(r'_{i+k})$	or $\mathrm{Im}(r'_{i+k}) > -2$
a_i or b_i = 1	if	$2 \geqslant \mathrm{Re}(r'_{i+k})$	or $\mathrm{Im}(r'_{i+k}) > 0$
a_i or b_i = 3	if	$\mathrm{Re}(r'_{i+k})$	or $\mathrm{Im}(r'_{i+k}) > 2$

Table 2.8 Threshold detection of a 16-point QAM signal s_i $(= a_i + jb_i)$ for the signal processor shown in Fig. 2.18. Re(.) and Im(.) are the real and imaginary parts of (.) respectively.

A linear equalizer has a serious weakness of not being able to equalize a baseband channel with a finite number of taps when one or more zeros of the z-transform of the channel sampled impulse response lie on the unit circuit in the z-plane.(A9)

2.6.2 Pure nonlinear equalizer

A pure nonlinear equalizer implemented as an arrangement of decision directed cancellation of intersymbol interference is now considered. The signal processor in Fig. 2.3 is now replaced by the pure nonlinear equalizer shown in Fig. 2.20. This equalizer is nonlinear because a threshold detector (being a nonlinear device) is included here.

The signals marked in Fig. 2.20 are those present at time $t = iT$. The sample value r_i at the input of the nonlinear equalizer is, from eqn. 2.19,

$$r_i = \sum_{h=0}^{g} s_{i-h} y_h + w_i = s_i y_0 + \sum_{h=1}^{g} s_{i-h} y_h + w_i \qquad (2.30)$$

where $\{s_i\}$ are the transmitted data symbols, $y_0, y_1, \cdots, y_g$ are the $(g+1)$ components of the channel sampled impulse response, and $\{w_i\}$ are the noise components at the output of the receiver filter shown in Fig. 2.3.

Assuming the correct detections of the data symbols $s_{i-1}, s_{i-2}, \cdots, s_{i-g}$, eqn. 2.30 and Fig. 2.20 show that the sample value at the input of the threshold detector is

$$r_i' = s_i + \frac{w_i}{y_0} \qquad (2.31)$$

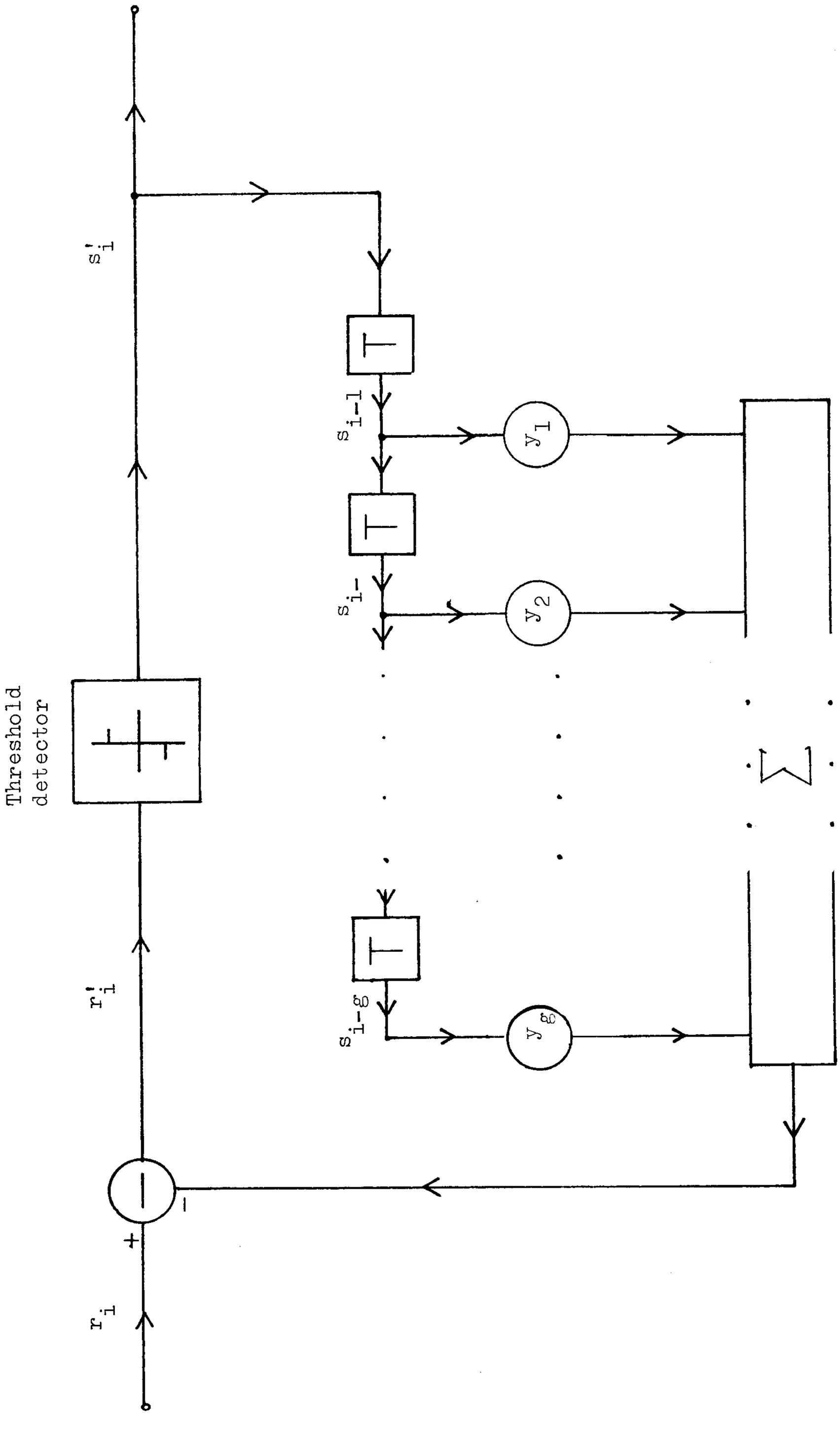


Fig. 2.20 A pure nonlinear equalizer.

where the intersymbol interference introduced by s_{i-1}, s_{i-2}, ·· , s_{i-g} have obviously been eliminated. It can be shown that the real and imaginary parts of the noise components $\left\{\frac{w_i}{y_0}\right\}$ (eqn. 2.31) are Gaussian random variables with zero mean and variance[A9]

$$\eta^2 = \frac{\sigma^2}{|y_0|^2} \qquad (2.32)$$

where $|y_0|$ is the modulus of the complex-valued quantity y_0, and σ^2 is the variance of the real and imaginary parts of the noise components $\{w_i\}$.

The detected value of the data symbol s_i is here determined according to Table 2.8 but with r'_{i+k} being replaced by r'_i (eqn. 2.31). The average probability of error in the detection of a_i or b_i is given by[A8]

$$\left(\frac{3}{2}\right) Q\left(\frac{1}{\eta}\right) = \left(\frac{3}{2}\right) Q\left(\frac{|y_0|}{\sigma}\right) \qquad (2.33)$$

where a_i and b_i are the real and imaginary parts of the data symbol s_i respectively, and $Q(.)$ is the well-known Q-function.

If, however, one or more of the previous data symbols s_{i-1}, s_{i-2}, ·· , s_{i-g} in eqn. 2.30 are incorrectly detected, then the intersymbol interference level is increased instead of being eliminated, leading to an increase in the probability of error in the detection of a_i or b_i. Consequently, errors tend to occur in bursts and the system suffers from error extension effect. At high signal to noise ratios, the error extension effect is likely to be less severe and eqn. 2.33 may be taken as a fair approximation to the average probability of error in the detection of a_i or b_i.[A8]

When all the zeros of the z-transform of the channel sampled impulse response lie inside the unit circle in the z-plane, y_0 is one of the larger components (in magnitude) of the channel sampled impulse response.[A9] In this case and at high signal to noise ratios, a pure nonlinear equalizer normally gives a better tolerance to noise than that of a linear equalizer.[A9] Conversely, when all zeros of the z-transform of the channel sampled impulse response lie outside the unit circle, then a linear equalizer normally gives a better tolerance to noise than that of the pure nonlinear equalizer.[A9]

2.6.3 Optimum nonlinear equalizer

Section 2.6.2 has indicated that at high signal to noise ratios, the tolerance to noise of a pure nonlinear equalizer is likely to be improved if the value of $|y_0|$ is larger, where y_0 is the first component of the channel sampled impulse response. It therefore follows that the tolerance to noise may be improved by inserting a linear filter at the input of the pure nonlinear equalizer so that the sampled impulse response of the channel and linear filter has a large first component. The signal processor (Fig. 2.3) now has the arrangement shown in Fig. 2.21.

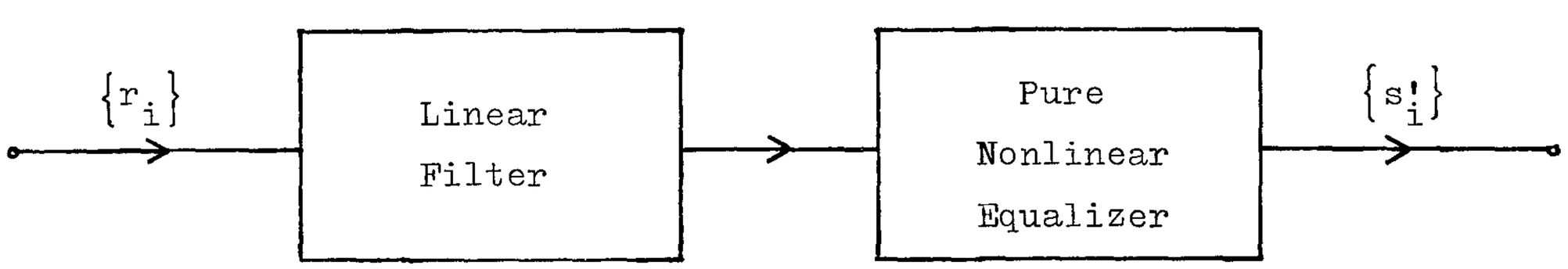


Fig. 2.21 A nonlinear equalizer implemented as a combination of a pure nonlinear equalizer (section 2.6.2) and a linear filter

For convenience, the arrangement shown in Fig. 2.21 will from now on be referred to as the 'nonlinear equalizer' in this thesis. Various techniques have been used to design the linear filter of this nonlinear equalizer.[A9] One of these techniques is to use the linear filter to force to zero the components preceeding that of the largest magnitude in the channel sampled impulse response. The second approach is to consider the z-transform of the channel sampled impulse response as consisting of two parts. That is

$$V(z) = V_i(z)\, V_o(z) \qquad (2.34)$$

where all the zeros (roots) of $V_i(z)$ are located on or inside the unit circle in the z-plane, and all the zeros of $V_o(z)$ are located outside the unit circle. The linear filter here is selected as a linear equalizer that equalizes $V_o(z)$. The third approach is to select the linear filter such that the average probability of error in the detection of the data symbols $\{s_i\}$ is minimized, assuming a high signal to noise ratio but neglecting the error extension effects. This is the optimum nonlinear equalizer and the more important properties of the linear filter used here will now be described.

Basically, the linear filter used in the optimum nonlinear equalizer is an all-pass network that does not change the amplitude distortion in the received signal but makes the sampled impulse response of the channel and filter 'minimum phase', thus concentrating the energy of each received signal element towards the start of that element.[A9] It is also shown in Ref. A9 that the noise components at the linear filter output have the same statistical properties as those at its input. That is, if the real and imaginary parts of the noise components $\{w_i\}$ at the input of the linear filter are statistically independent

Gaussian random variables with zero mean and variance σ^2, then the real and imaginary parts of the noise components $\{w_i'\}$ at the output of the linear filter are also statistically independent Gaussian random variables with zero mean and variance η^2, $\eta^2 = \sigma^2$ when the tap gains of the linear filter, $f_0, f_1, \cdots, f_p$, are such that

$$\sum_{h=0}^{p} |f_h|^2 = 1 \qquad (2.35)$$

Furthermore, it can be shown that the z-transform of the linear filter having the properties just described is given by[A9]

$$F(z) = \frac{V_o'(z)}{V_o(z)} \qquad (2.36)$$

where $V_o(z)$ is as defined in eqn. 2.34, and $V_o'(z)$ is obtained from $V_o(z)$ by reversing the order of the complex conjugates of the coefficients of $V_o(z)$. Thus if the z-transform of $V_o(z)$ is given by

$$V_o(z) = v_0 + v_1 z^{-1} + \cdots + v_k z^{-k} \qquad (2.37)$$

then the z-transform of $V_o'(z)$ is given by

$$V_o'(z) = v_k^* + v_{k-1}^* z^{-1} + \cdots + v_0^* z^{-k} \qquad (2.38)$$

where v_i^* is the complex conjugate of v_i, and the zeros (or roots) of $V_o'(z)$ are the complex conjugates of the reciprocals of the zeros of $V_o(z)$.[A9] It now follows that the linear filter here (eqn. 2.36) effectively replaces all zeros of the z-transform of the channel that lie outside the unit circle in the z-plane, by the complex conjugates of their reciprocals, all remaining zeros being unchanged. All zeros of the z-transform of the channel and filter now lie inside the unit circle in the z-plane and so the first component of the sampled impulse response of the channel and filter is now one of the larger components in this response.

Since the noise components at the linear filter output have the same statistical properties as those at its input, the tolerance to noise of the optimum nonlinear equalizer can be evaluated by using a modified arrangement of Fig. 2.3 that includes the linear filter as part of the baseband channel and takes just the pure nonlinear equalizer as the signal processor. This arrangement is very convenient and is in fact used in the computer simulation tests in this thesis to determine the tolerance to noise of the optimum nonlinear equalizer. Thus, for the given data-transmission system of Fig. 2.3, instead of determining the tolerance to noise of the optimum nonlinear equalizer operating over the channel sampled impulse response (Tables 2.4 and 2.5), the computer simulation tests in chapter 5 now determine the tolerance to noise of the pure nonlinear equalizer (section 2.6.2) operating over the sampled impulse response of the channel and filter (Tables 2.6 and 2.7). The sampled impulse responses shown in Tables 2.6 and 2.7 are, of course, obtained from those shown in Tables 2.4 and 2.5 by replacing all zeros of their z-transform that lie outside the unit circle in the z-plane by the complex conjugates of their reciprocals, leaving the remaining zeros unchanged. Consequently, at high signal to noise ratios and neglecting the error extension effect, the average probability of error in the detection of the real or imaginary parts of the data symbols $\{s_i\}$ is here similar to that given by eqn. 2.33 and is approximately given by

$$\left(\frac{3}{2}\right) Q\left(\frac{|y'_0|}{\sigma}\right) \qquad (2.39)$$

where y'_0 is the first component of the sampled impulse response of the channel and linear filter, and σ^2 is the variance of the real and imaginary parts of the noise components in the received sample values.

It is also shown in Ref. A9 that the linear filter used in the optimum nonlinear equalizer here is the same as the linear feedforward transversal filter that forms the first part of the conventional decision feedback nonlinear equalizer, where the linear filter is adjusted to minimize the mean-square error (and hence maximize the signal to noise ratio) in its output signal, assuming the exact equalization of the channel.

2.7 Maximum Likelihood Detection Process and the Viterbi Algorithm Detection Process

One weakness of the signal processor implemented as the optimum nonlinear equalizer previously described is that only a portion of the received signal energy (corresponding to the energy carried by the first component y_0' of the sampled impulse response of the channel and linear filter) is used in the detection of that signal. Clearly, the truely optimum detection process that minimizes the detection error rate should make use of all the available energy of the received signal elements. This optimum detection process is now described as below.

It is assumed that transmission starts at time $t = T$, and $s_i = 0$ for $i \leq 0$, for the data-transmission system shown in Fig. 2.3. The ith received sample value at the output of the sampler (Fig. 2.3) is, from eqn. 2.19, given by

$$r_i = q_i + w_i \tag{2.40}$$

where
$$q_i = \sum_{h=0}^{g} s_{i-h} y_h \tag{2.41}$$

and $s_{i-h} = 0$ for $i-h \leq 0$. The components $\{r_i\}$, $\{q_i\}$, $\{w_i\}$, $\{s_i\}$ and

$\{y_h\}$ are, in general, complex-valued quantities and are as defined before. Thus, $y_0, y_1, \cdots, y_g$ are the (g+1) components of the channel sampled impulse response, $\{s_i\}$ are the transmitted data-symbol values, and $\{w_i\}$ are the noise components whose real and imaginary parts are statistically independent Gaussian random variables with zero mean and variance σ^2.

If K is the total number of transmitted data symbols, then all the received sample values $\{r_i\}$ for the whole transmission period (KT seconds) can be represented by a K-component vector R_K, where

$$R_K = \begin{bmatrix} r_1 & r_2 & \cdots & r_K \end{bmatrix} \tag{2.42}$$

Similarly, the corresponding transmitted data symbols $\{s_i\}$, the received signal sample values $\{q_i\}$, and the noise components $\{w_i\}$ can be represented by the following three K-component vectors respectively.

$$S_K = \begin{bmatrix} s_1 & s_2 & \cdots & s_K \end{bmatrix} \tag{2.43}$$

$$Q_K = \begin{bmatrix} q_1 & q_2 & \cdots & q_K \end{bmatrix} \tag{2.44}$$

$$W_K = \begin{bmatrix} w_1 & w_2 & \cdots & w_K \end{bmatrix} \tag{2.45}$$

Eqn. 2.40 now becomes

$$R_K = Q_K + W_K \tag{2.46}$$

It can be seen from eqns. 2.41, 2.43, and 2.44 that S_K is uniquely determined by Q_K. When the 16-point QAM signals are considered, there are, in general, 16^K different possible vectors of S_K or Q_K. The signal processor in Fig. 2.3 is assumed to have prior knowledges of all the 16^K possible vectors of S_K and their corresponding a priori probabilities.

2.7.1 Maximum likelihood detection process

Thus, all prior knowledge in the data-transmission system is used in the optimum detection process (that minimizes the probability of detection error) to determine the detected data-symbol vector S'_K, of S_K (eqn. 2.43). In order to minimize the probability of error in the detection process, it is necessary to maximize the probability of correct detection $P(C)$,[A9] where

$$P(C) = \int_{-\infty}^{\infty} P(C|R_K)\, f(R_K)\, dR_K \qquad (2.47)$$

$P(C|R_K)$ is the conditional probability of correct detection of the data-symbol vector S_K given the received vector R_K (eqn. 2.42), and $f(R_K)$ is the probability density function of R_K. Since $f(R_K)$ is non-negative by the nature of a probability density function, therefore $P(C)$ is maximized by maximizing the value of $P(C|R_K)$ for every possible vector of R_K. Furthermore, for any given vector R_K, the value of $P(C|R_K)$ is maximized by taking, as the detected vector, the possible vector of S_K, for which the value of $P(S_K|R_K)$ is the maximum, where $P(S_K|R_K)$ is the conditicnal probability of S_K given R_K and is thus the a posteriori probability of S_K. The optimum detection process is now reduced to the maximum a posteriori (M.A.P.) detection process.

Since there is a one to one mapping between S_K and Q_K (eqn. 2.44), therefore,

$$P(S_K|R_K) = P(Q_K|R_K) \qquad (2.48)$$

Moreover, by Bayes' theorem,

$$P(Q_K|R_K) = \frac{P(Q_K)}{f(R_K)}\, f(R_K|Q_K) \qquad (2.49)$$

where $P(Q_K)$ is the a priori probability of Q_K, $f(R_K)$ is the probability density function of R_K, and $f(R_K|Q_K)$ is the conditional probability density function of R_K given Q_K. For a given received vector R_K, $f(R_K)$ is independent of Q_K. The detected data-symbol vector S'_K that maximizes the quantity $P(S_K|R_K)$ is therefore also the vector that maximizes the quantity $P(Q_K)f(R_K|Q_K)$. In the particular case where all the 16^K possible vectors $\{S_K\}$ are equally likely to be transmitted,

$$P(S_K) = P(Q_K) = 16^{-K} \qquad (2.50)$$

and the optimum detection process now reduces to one that selects, as the detected vector, the possible vector of S_K for which the corresponding value of $f(R_K|Q_K)$ is the maximum. $f(R_K|Q_K)$ is often known as the likelihood function of Q_K and the optimum detection process, under the assumed conditions, is thus the maximum likelihood detection process.

The likelihood function $f(R_K|Q_K)$ is in fact the conditional joint probability density function of the random variable with sample values $r_1, r_2, \cdot\cdot, r_K$, (eqn. 2.42) given the values of $q_1, q_2, \cdot\cdot, q_K$, (eqn. 2.44). That is

$$f(R_K|Q_K) = f(r_1, r_2, \cdot\cdot, r_K | q_1, q_2, \cdot\cdot, q_K) \qquad (2.51)$$

Since the sample values $\{r_i\}$ and $\{q_i\}$ are complex-valued quantities, where

$$r_i = r_{i,1} + j\, r_{i,2} \qquad (2.52)$$

$$q_i = q_{i,1} + j\, q_{i,2} \qquad (2.53)$$

for $i = 1,2,\cdot\cdot,K$ and $j = \sqrt{-1}$, therefore eqn. 2.51 now becomes

$$f(R_K|Q_K) = f(\ r_{1,1},r_{1,2},r_{2,1},\cdot\cdot,r_{K,1},r_{K,2}|\ q_{1,1},q_{1,2},q_{2,1},\cdot\cdot,q_{K,1},q_{K,2}\ ) \qquad (2.54)$$

Since the real and imaginary parts of the data symbols $\{s_i\}$ are statistically independent (section 2.4), it follows from eqns. 2.40 and 2.41 that the real and imaginary parts of the sample values $\{q_i\}$ and $\{r_i\}$ must also be statistically independent, so that eqn. 2.54 may now be simplified to

$$f(R_K|Q_K) = f(r_{1,1}|q_{1,1})\ f(r_{1,2}|q_{1,2})\ f(r_{2,1}|q_{2,1}) \cdot\cdot\ f(r_{K,1}|q_{K,1})\ f(r_{K,2}|q_{K,2}) \qquad (2.55)$$

Furthermore, since the real and imaginary parts of the noise components $\{w_i\}$ in eqn. 2.40 are statistically independent Gaussian random variables with zero mean and variance σ^2 (section 2.4), therefore for a given value of $q_{i,h}$, $r_{i,h}$ is a sample value of the Gaussian random variable with mean $q_{i,h}$ and variance σ^2. That is

$$f(r_{i,h}|q_{i,h}) = \frac{1}{\sqrt{2\pi(\sigma)^2}}\ \exp\left(\frac{-(r_{i,h}-\ q_{i,h})^2}{2(\sigma)^2}\right) \qquad (2.56)$$

for $h = 1,2$ and $i = 1,2,\cdot\cdot,K$. Substituting eqn. 2.56 into eqn. 2.55 and after some rearranging of terms, we arrive at

$$f(R_K|Q_K) = \frac{1}{(2\pi(\sigma)^2)^K}\ \exp\left(\frac{\|R_K-\ Q_K\|^2}{-\ 2(\sigma)^2}\right) \qquad (2.57)$$

and
$$\|R_K-\ Q_K\|^2 = \sum_{i=1}^{K}|r_i-\ q_i|^2 = \sum_{i=1}^{K}\left((r_{i,1}-\ q_{i,1})^2+(r_{i,2}-\ q_{i,2})^2\right) \qquad (2.58)$$

where $\|R_K-\ Q_K\|$ is the unitary distance between the received vector R_K and the received signal vector Q_K. It can be seen from eqn. 2.57 that the value of the likelihood function $f(R_K|Q_K)$ is maximized when

the unitary distance $\|R_K - Q_K\|$ is minimized. Consequently, the maximum likelihood detection process, under the assumed conditions, selects as the detected vector the possible vector of S_K for which the corresponding signal vector Q_K is closest (in terms of the unitary distance) to the received vector R_K.

Clearly, an error occurs in the detection of the data-symbol vector S_K when the received signal vector Q_K is not the possible vector of Q_K that is closest to the received vector R_K. This necessarily involves one or more errors in the detected data-symbol values $\{s_i'\}$. It is, however, difficult to evaluate exactly the average probability of error in the detection of the real or imaginary parts of the data symbols $\{s_i\}$, but an approximate upper bound to this average probability of error at high signal to noise ratios may be obtained as $Q(\frac{d_{min}}{\sigma})$, (A9) where $Q(.)$ is the well-known Q-function, σ^2 is the variance of the real and imaginary parts of the noise components $\{w_i\}$, and d_{min} is the minimum unitary distance between any two possible received signal vectors $\{Q_K\}$. The real-valued quantity d_{min} therefore appears to be the determining factor of the tolerance to noise of the maximum likelihood detection process at high signal to noise ratios.

2.7.2 Viterbi algorithm detection process

The maximum likelihood detection process described in section 2.7.1 is obviously too complex to be of any practical use. In particular, when the 16-point QAM signals are used, all the 16^K possible data-symbol vectors $\{S_K\}$ must be stored at the receiver and a total of 16^K unitary distance measurements are required at the end of the transmission when all the data symbols $\{s_i\}$ are detected simultaneously. K is here the total number of transmitted data symbols and is usually a very

large value in practice.

Fortunately, there are superfluous operations in the maximum likelihood detection process, in the sense that not all the 16^K possible vectors $\{S_K\}$ are actually required to be considered in the detection process. A sequential detection process based on the Viterbi algorithm is known to achieve the maximum likelihood detection while getting rid of the superfluous operations.(A10,C10) The basic principle and operation of the Viterbi algorithm detection process will now be outlined and discussed as below.

The Viterbi algorithm was first introduced to decode the convolutionally coded signals.(A10,D5,D8,D16) Since the operation of the convolutional encoder is very similar to the operation of convolving the transmitted sequence of $\{s_i\}$ with the channel sampled impulse response (eqn. 2.19), the Viterbi algorithm can be used to detect messages that are transmitted over a time-dispersive channel. Thus, the Viterbi algorithm detector operates as follows. Just before the receipt of the sample r_i at the detector input (Fig. 2.3), the Viterbi algorithm detector holds in store m_v (i-1)-component vectors $\{Z_{i-1}\}$,

$$Z_{i-1} = [x_1 \; x_2 \cdot\cdot x_{i-1}] \qquad (2.59)$$

where x_h has one of the 16 possible values of the complex-valued data symbol s_h. m_v is here a suitable value to be discussed shortly. Associated with each stored vector Z_{i-1} is stored the corresponding value of cost C_{i-1} which will shortly be discussed. On receiving the sample r_i, each of the m_v stored vectors $\{Z_{i-1}\}$ is expanded into 16 vectors $\{Z_i\}$. The first (i-1) components of Z_i are as in the original vector Z_{i-1} and the last component x_i has the 16 different possible values of s_i. The detector then evaluates the cost C_i

associated with each expanded vector Z_i. The cost C_i is here defined as

$$C_i = c_1 + c_2 + \cdots + c_i \tag{2.60}$$

where $$c_j = \left| r_j - \sum_{h=0}^{g} x_{j-h} y_h \right|^2 \tag{2.61}$$

for all positive integer values of j, and $x_{j-h} = 0$ for $j-h \leq 0$. The complex-valued quantities $y_0, y_1, \cdots, y_g$ are of course the g+1 components of the channel sampled impulse response. Thus, the cost associated with each expanded vector Z_i is here evaluated as

$$C_i = C_{i-1} + c_i \tag{2.63}$$

where the value of C_{i-1} has already been evaluated previously and the value of c_i can be evaluated by using eqn. 2.61.

It can be seen from eqns. 2.41, 2.58, 2.60, and 2.61 that, the cost C_i is actually the square of the unitary distance between the received vector R_i (eqn. 2.42) and the possible vector of Q_i (eqn. 2.44) corresponding to the possible vector of S_i (eqn. 2.43) that has the same components as those of the expanded vector Z_i (eqn. 2.59). Thus, in the original maximum likelihood detection process, all the expanded vectors $\{Z_i\}$ are stored and used for the fcllowing signal processing until the end of the transmission when the vector associated with the minimum cost is taken as the detected data-symbol vector. In the Viterbi algorithm detection process, however, only m_v of the $16m_v$ expanded vectors $\{Z_i\}$ are selected and stored, while the rest, being redundant, are discarded from further consideration. The value of m_v, being the number of stored vectors used in the Viterbi algorithm detection process, is now derived as

follows. It can be seen from eqn. 2.61 that the values of c_{i+1}, c_{i+2}, $\cdots$ are independent of the values of x_1, x_2, $\cdots$, x_{i-g}. Furthermore, all those vectors of $\{Z_K\}$ (K being the total number of transmitted data symbols) that are originated from those vectors of $\{Z_i\}$ having the same set of values for the last g components x_{i-g+1}, x_{i-g+2}, $\cdots$, x_i, must have the same set of values for c_{i+1}, c_{i+2}, $\cdots$, c_K so long as these $\{Z_K\}$ have the same set of values for x_{i+1}, x_{i+2}, $\cdots$, x_K. It now follows from eqn. 2.60 that, any of these vectors $\{Z_K\}$ (which have the same set of values for c_{i+1}, c_{i+2}, $\cdots$, c_K) that has a larger value of C_i must also have a larger value of C_K, bearing in mind that C_i is the cost of Z_i. However, only the vector Z_K associated with the minimum cost C_K will be taken as the detected data-symbol vector in the maximum likelihood detection process. It therefore follows that for a given set of vectors $\{Z_i\}$ that have the same set of last g components x_{i-g+1}, x_{i-g+2}, $\cdots$, x_i, all vectors $\{Z_K\}$ originating from these $\{Z_i\}$, except the Z_i associated with the smallest cost C_i for this set, will never be selected as the detected data-symbol vector and can thus be discard from further consideration. Consequently, for those vectors of $\{Z_i\}$ that have the same set of last g components, it is necessary to store just the vector that has the smallest cost C_i. For a 16-point QAM signal, there are altogether 16^g possible combinations for the g components x_{i-g+1}, x_{i-g+2}, $\cdots$, x_i of Z_i, and the total number of vectors $\{Z_i\}$ required to be stored here is therefore

$$m_v = 16^g \tag{2.63}$$

where g+1 is the number of components in the channel sampled impulse response.

Thus, the Viterbi algorithm detection process selects from the $16m_v$ expanded vectors $\{Z_i\}$, the m_v vectors associated with the smallest costs $\{C_i\}$, subject to the constraint that these m_v selected vectors have the m_v possible combinations for the last g components of Z_i. The detection process continues in this way, so that each time after a sample is received, $16m_v$ costs measurements are carried out for the $16m_v$ expanded vectors, and m_v vectors are then selected in the way described above. At the end of the transmission, the stored vector Z_K associated with the minimum cost C_K is taken as the detected data-symbol vector which is obviously the same as that obtained in the maximum likelihood detection process. The maximum likelihood detection process can therefore be implemented by using the Viterbi algorithm detection process just described without any loss in tolerance to noise.

In practice, it is always desirable to detect the transmitted message with as little delay as possible. To do this, the data symbol s_{i-n} may be detected as the value of x_{i-n} in the stored vector Z_i (eqn. 2.59) having the minimum cost C_i. In this case, the delay in detection is n sampling intervals, and when $n > 3g+1$, the reduction in tolerance to noise in relation to that of the maximum likelihood detection process (where $n = K$) has been known to be negligible.[A9,A10] Clearly, after the detection of s_{i-n}, the component x_{i-n} is no longer required in the detection process and can thus be discarded from the m_v stored vectors $\{Z_i\}$. Similarly, all the components x_1, x_2, .. , x_{i-n-1} are also not required to be stored here. Consequently, only the last n components of the m_v selected vectors are required to be stored here. Thus, just before the receipt of r_i (Fig. 2.3), the detector now holds in store m_v n-component vectors $\{X_{i-1}\}$, in

place of the m_v (i-1)-component vectors $\{Z_{i-1}\}$ given in eqn. 2.59, where

$$X_{i-1} = \left[x_{i-n}\ x_{i-n+1}\ \cdots\ x_{i-1}\right] \tag{2.64}$$

After the receipt of r_i, each of the m_v stored vectors $\{X_{i-1}\}$ is expanded into 16 (n+1)-component vectors having the 16 possible values of x_i. $16m_v$ cost measurements (eqns. 2.60, 2.61, and 2.62) are then carried out for the $16m_v$ expanded vectors, and the appropriate m_v vectors are then selected as previously described. The detected data symbol s'_{i-n} is then taken as the value of x_{i-n} in the selected vector associated with the minimum cost. The component x_{i-n} is then discarded from each of the m_v selected vectors to give the m_v stored vectors $\{X_i\}$ (eqn. 2.64). The detection process then proceeds to operate on the next received sample value r_{i+1} in the same way as is described before.

CHAPTER 3

ITERATIVE DETECTION PROCESSES FOR BINARY BASEBAND SIGNALS

3.1 Introduction

Recently, it has been shown that the tolerance to additive white Gaussian noise of a detection process based on some modifications of the Gauss-Seidel iterative process[E1] can be made to approach that of the optimum detection process.[A9,C29,C30,C36] The iterative detection process here basically involves a sequence of identical or similar operations which are performed successively by a fairly simple piece of equipment. This chapter is concerned with the study of some further developments of this type of iterative detection process. The aim here is to further reduce the equipment complexity and the number of operations per data symbol involved in the detection process in achieving the best tolerance to additive white Gaussian noise.

3.2 Basic Assumptions

The basic model of the data-transmission system here is the synchronous serial binary baseband data-transmission system described in section 2.1. Thus, from section 2.1, the sample value at the output of the receiver filter in Fig. 2.1, at time $t = iT$, is

$$r_i = \sum_{h=0}^{g} s_{i-h} y_h + w_i \qquad (3.1)$$

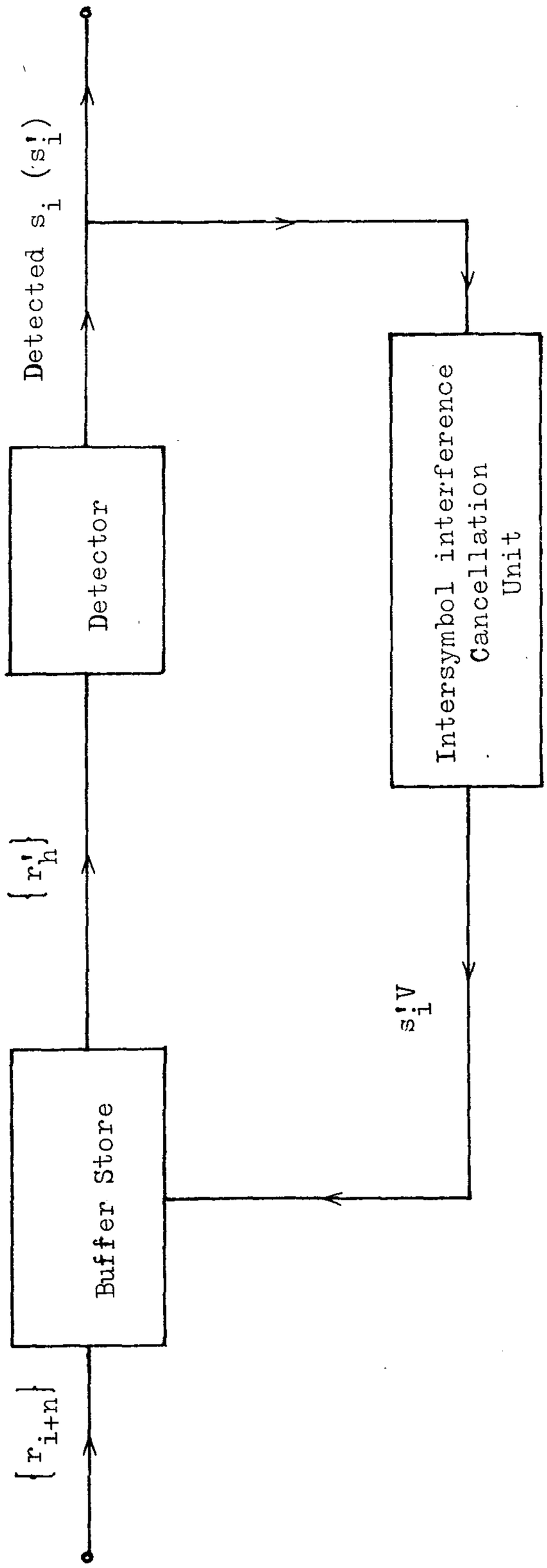
where the $\{s_i\}$ are the data symbols whose values are to be detected at the receiver, $y_0, y_1, \cdots, y_g$ are the g+1 components of the

sampled impulse response of the baseband channel, and the $\{w_i\}$ are the noise components here. It is assumed that the data symbols $\{s_i\}$ are statistically independent and equally likely to have any of the two binary values 1 and -1, and the noise components $\{w_i\}$ are the statistically independent Gaussian random variables with zero mean and a fixed variance σ^2 where σ^2 is the two-sided power spectral density of the noise added at the output of the transmission path shown in Fig. 2.1. It is also assumed that the channel sampled impulse response is known and time-invariant. The signal processor in Fig. 2.1 is here implemented as a detection process that employs the arrangement of detection and intersymbol interference cancellation shown in Fig. 3.1.

Thus, in the detection of the data symbol s_i, the buffer store in Fig. 3.1 holds the n sample values $r'_1, r'_2, \cdots, r'_n$, assuming for the moment that $n > g$ where g+1 is the number of components of the channel sampled impulse response. These n sample values are obtained from the n successive received sample values $r_i, r_{i+1}, \cdots, r_{i+n-1}$ (eqn. 3.1), by removing from them detected values of all components involving data symbols whose final detected values have already been determined. That is

$$\begin{aligned}
r'_1 &= r_i - s'_{i-1}y_1 - s'_{i-2}y_2 - \cdots\cdots - s'_{i-g}y_g \\
r'_2 &= r_{i+1} - s'_{i-1}y_2 - s'_{i-2}y_3 - \cdots - s'_{i-g+1}y_g \\
&\vdots \\
r'_n &= r_{i+n-1}
\end{aligned} \tag{3.2}$$

where the $\{s'_j\}$ are the final detected values of the data symbols $\{s_j\}$



Signal processor using the arrangement of Detection and Intersymbol Interference Cancellation

Fig. 3.1

The detector (Fig. 3.1) operates on its n input sample values given by eqn. 3.2 to give at its output the detected value of s_i. The detector here is implemented by using some iterative processes to be developed and studied in this chapter. Having determined the detected value s_i', of s_i, the intersymbol interference cancellation unit (Fig. 3.1) then generates the (g+1)-component vector $s_i'V$,

$$s_i'V = \left[s_i'y_0 \quad s_i'y_1 \quad \cdots \quad s_i'y_g \right] \tag{3.3}$$

where V is the channel sampled impulse response given by eqn. 2.3. The components of $s_i'V$ are then removed from the n sample values held in the buffer store (Fig. 3.1), so that the first (g+1) sample values held in the buffer store now become $r_1' - s_i'y_0$, $r_2' - s_i'y_1$, $\cdots$, $r_g' - s_i'y_g$ respectively. The first sample value $r_1' - s_i'y_0$ is then discarded from the buffer store and the remaining (n-1) sample values are shifted one place to the front to become the new sample values r_1', r_2', $\cdots$, r_{n-1}' respectively. The next received sample value r_{i+n} is then moved into the buffer store to become the new sample value r_n'. The detected value of the next data symbol s_{i+1} can now be determined from the new n sample values held in the buffer store in exactly the same way as is described before for the detection of s_i, and the detection process continues in this way.

In the detection of s_i, the detector here assumes that the g previous data symbols s_{i-1}, s_{i-2}, $\cdots$, s_{i-g} have been detected correctly so that the components involving these data symbols are removed from the n received sample values in eqn. 3.2. It can be seen from eqns. 3.1 and 3.2 that the n sample values held in the buffer store now become

$$
\begin{aligned}
r'_1 &= s_i y_0 + w_i \\
r'_2 &= s_{i+1} y_0 + s_i y_1 + w_{i+1} \\
&\vdots \\
r'_n &= s_{i+n-1} y_0 + s_{i+n-2} y_1 + \cdots + s_{i+n-g-1} y_g + w_{i+n-1}
\end{aligned} \tag{3.4}
$$

where $\{w_h\}$ are the noise components defined in eqn. 3.1. If, however, one or more of the g previous data symbols $s_{i-1}, s_{i-2}, \cdots, s_{i-g}$ have been detected incorrectly, then the cancellation of these data symbols doubles instead of removes their intersymbol interference in the n sample values $r'_1, r'_2, \cdots, r'_n$ (eqn. 3.2), thus increasing the probability of error in the detection of s_i. Consequently, errors in the detected data symbols tend to occur in bursts and the detection process here (Fig. 3.1) suffers from error-extension effects.

To start the process of detection and intersymbol interference cancellation, a known sequence of more than g data symbols $\{s_i\}$ is transmitted, and the components associated with these data symbols are automatically cancelled in the buffer store, without the detection of these data symbols.

If $n < g+1$, the buffer store holds g+1 sample values instead of n, and the detection process operates on only the first n of these. The value of n is assumed to be 8 in this chapter.

Having described the process of detection and intersymbol interference cancellation, the remaining part of this chapter is mainly concerned with the study of the various iterative processes to be used in the detector of Fig. 3.1. Consequently, only the process of detecting the data symbol s_i from the n sample values $r'_1, r'_2, \cdots, r'_n$

will be described in each of the systems studied and developed in the following sections of this chapter. To simplify the nomenclature in the descriptions of the operations of the various systems here, rename s_i as s_1, s_{i+1} as s_2, and so on, and let S be the n-component vector

$$S = \begin{bmatrix} s_1 & s_2 & \cdot\cdot & s_n \end{bmatrix} \tag{3.5}$$

Similarly, rename w_i as w_1, w_{i+1} as w_2, and so on, and let W be the n-component vector

$$W = \begin{bmatrix} w_1 & w_2 & \cdot\cdot & w_n \end{bmatrix} \tag{3.6}$$

Thus, if the n sample values held in the buffer store are given by the n-component vector

$$R' = \begin{bmatrix} r'_1 & r'_2 & \cdot\cdot & r'_n \end{bmatrix} \tag{3.7}$$

then eqn. 3.4 now reduces to

$$R' = SY + W \tag{3.8}$$

where Y is the $n \times n$ matrix whose ith row is given by the n-component vector

$$Y_i = \begin{bmatrix} \overbrace{0 \cdots 0}^{i-1} & y_0 & y_1 & \cdots & y_{n-i} \end{bmatrix} \tag{3.9}$$

and $y_i = 0$ for $i < 0$, $i > g$. In eqn. 3.8, SY is the n-component signal vector and W is the n-component noise vector. Since Y is here an upper triangular matrix with a non-zero main diagonal ($y_0 \neq 0$), it is non-singular with rank n. The n vectors $Y_1, Y_2, \cdot\cdot, Y_n$ are therefore linearly independent and the data-symbol vector S may be uniquely determined from SY.

Thus, the detector in Fig. 3.1 now operates on the n-component vector R' (eqn. 3.7) to give the detected vector S', of S (eqn. 3.5), but only the first component of S' is here taken as the detected value s_1' of the data symbol s_1 (or s_i in Fig. 3.1). The intersymbol interference cancellation unit (Fig. 3.1) then generates the (g+1)-component vector

$$s_1'V = \left[s_1'y_0 \quad s_1'y_1 \quad \cdots \quad s_1'y_g \right] \tag{3.10}$$

instead of $s_i'V$ given by eqn. 3.3. The components of $s_1'V$ are then removed (by subtraction) from the first g+1 components of R'. The resultant sample values are then moved one place to the left to give the new vector R', whose extreme right-hand component is the next received sample value r_{i+n}.

Various arrangements of the detection process for s_1 from R' have been studied and are referred to as the various systems described in the following sections of this chapter. The approach of study here is to include deliberately some of the poorer but more fundamental systems so as to clarify the operation of the more promising systems developed in this chapter. Except where otherwise stated, all results are obtained by using computer simulation technique. The sampled impulse responses of the channels to be used in this chapter are those given in Table 2.1. In order to avoid using excessive computing time, only channel A in Table 2.1 is used for the initial computer simulation tests to determine the tolerances to noise of the various systems studied and developed here. The more promising systems are then further tested on channels B and C in Table 2.1.

3.3 Optimum System

The optimum system here is defined as the system that minimizes the probability of error in the detection of the data-symbol vector S (eqn. 3.5) from the vector R' (eqn. 3.7), assuming that R' is as given by eqn. 3.8. It is shown in Ref. A9 that, under the various conditions assumed here, this optimum system selects as the detected data-symbol vector S', the possible vector of S for which the Euclidean distance between the corresponding n-component vector SY and the n-component vector R' is the minimum. This Euclidean distance is here defined as

$$\|R' - SY\| = \sqrt{\sum_{h=1}^{n} (r'_h - q_h)^2} \qquad (3.11)$$

where

$$q_h = \sum_{k=1}^{n} s_k y_{h-k} \qquad (3.12)$$

for $h = 1, 2, \cdots, n$ and $y_j = 0$ for $j > g$. The $\{s_i\}$ and $\{r'_i\}$ here are as defined in eqns. 3.5 and 3.7 respectively, and $y_0, y_1, \cdots, y_g$ are, of course, the g+1 components of the channel sampled impulse response.

Thus, in the detection process for s_1 from R', the optimum system here evaluates the distance (eqn. 3.11) associated with each possible vector of S. The detected data-symbol vector S', of S, is then taken as the possible vector of S associated with the minimum distance. The first component of S' is then accepted as the detected value of s_1.

It is interesting to note that, when n = 1, the arrangement of Fig. 3.1 with the optimum system just described becomes the pure nonlinear equalizer described in section 2.6.2.

Unfortunately, this optimum system has the serious weakness that it requires 2^n distance measurements to determine the detected value of s_1, and this becomes excessive when $n > 10$. Nevertheless, it is useful to evaluate the performance of this optimum system as a comparison with the simpler systems to be developed in the later sections of this chapter.

3.4 Gauss-Seidel Iterative Process

It has been shown[A9] that a linear estimate of the data-symbol vector S (eqn. 3.5) can be obtained from the vector R' (eqn. 3.8) by means of the Gauss-Seidel iterative process, whose method of operation can be explained with reference to Fig. 3.2. The n-component vector

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \tag{3.13}$$

in Fig. 3.2 is initially set to zero, and the n-component vector XY is subtracted from R' to give the vector R' - XY, where Y is the $n \times n$ matrix defined in eqn. 3.9. The square marked Y^T is an $n \times n$ network that performs the linear transformation Y^T on the input vector R' - XY, where Y^T is the transpose of Y. Thus, the n input terminals of the network hold the n components of the vector R' - XY, and the n output terminals hold the n components of the vector

$$E = \begin{bmatrix} e_1 & e_2 & \cdots & e_n \end{bmatrix} \tag{3.14}$$

where $$E = (R' - XY)Y^T \tag{3.15}$$

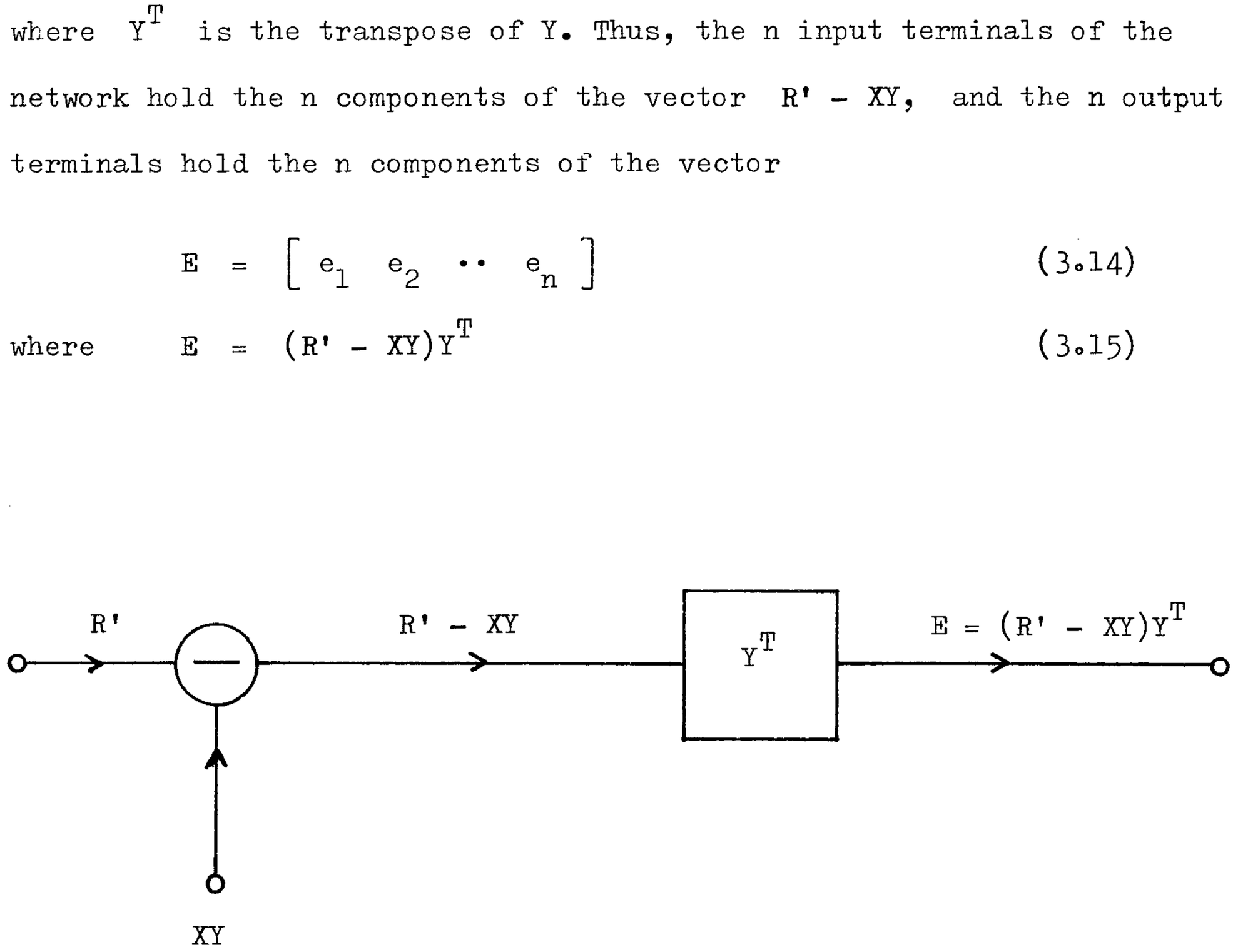


Fig. 3.2 Linear estimation process for S from R' using the Gauss-Seidel iterative process. X is the estimate of S.

The iterative process now operates as follows. x_1 is adjusted to set e_1 to zero. x_2 is then adjusted to set e_2 to zero, and the process continues in this way with the sequential adjustment of x_3 to x_n. The cycle of adjustments in the $\{x_i\}$ is now repeated in the same order many times until eventually $E \simeq 0$, when

$$(R' - XY)Y^T \simeq 0 \tag{3.16}$$

so that

$$X \simeq R'Y^T(YY^T)^{-1} = R'Y^{-1} \tag{3.17}$$

which is the maximum likelihood estimate of S.[A9]

The amount of adjustment required in the value of x_h to set e_h to zero for $h = 1,2,\cdot\cdot,n$ can be derived as follow. It can be seen from eqns. 3.9, 3.13, 3.14, and 3.15 that

$$e_h = (R' - XY)Y_h^T \tag{3.18}$$

$$\text{or} \quad e_h = (R' - x_1Y_1 - x_2Y_2 - \cdot\cdot - x_hY_h - \cdot\cdot - x_nY_n)Y_h^T \tag{3.19}$$

for $h = 1,2,\cdot\cdot,n$. Thus, if the increment in x_h is Δx_h so that the new (updated) value of e_h is zero, then from eqn. 3.19, the updated value of e_h is

$$\begin{aligned} 0 &= (R' - x_1Y_1 - \cdot\cdot - (x_h+\Delta x_h)Y_h - \cdot\cdot - x_nY_n)Y_h^T \\ &= (R' - XY)Y_h^T - \Delta x_h Y_h Y_h^T \\ &= e_h - \Delta x_h \|Y_h\|^2 \end{aligned} \tag{3.20}$$

and so

$$\Delta x_h = \frac{e_h}{\|Y_h\|^2} \tag{3.21}$$

where the $\{x_j\}$, $\{Y_j\}$, and e_h here are referred to those values before

the value of x_h is updated. The amount of adjustment required in x_h to set the value of e_h to zero can thus be obtained from the value of e_h using eqn. 3.21.

A more detailed diagram of the Gauss-Seidel iterative process is shown in Fig. 3.3. Thus, in Fig. 3.3, the control unit selects the value of h so that the vector $R' - XY$ is fed to the input of the network Y_h^T, where Y_h^T is a filter matched to Y_h. Having evaluated the value of e_h at the output of the network Y_h^T, the value of x_h and hence the vector $R' - XY$ are then updated from this value of e_h. This completes the adjustment of x_h and the control unit then selects the next value of h, and the operation continues as before.

Each of the systems to be developed and studied in this chapter uses an iterative process that is a further development of the Gauss-Seidel iterative process just described. The prior knowledge of the two possible values of the data symbol s_h is used here so that x_h may have any of the two possible values of s_h or 0 at all times during the iterative process. The value of x_h is only equal to 0 at the beginning of the iterative process when no decision is made for the value of x_h.

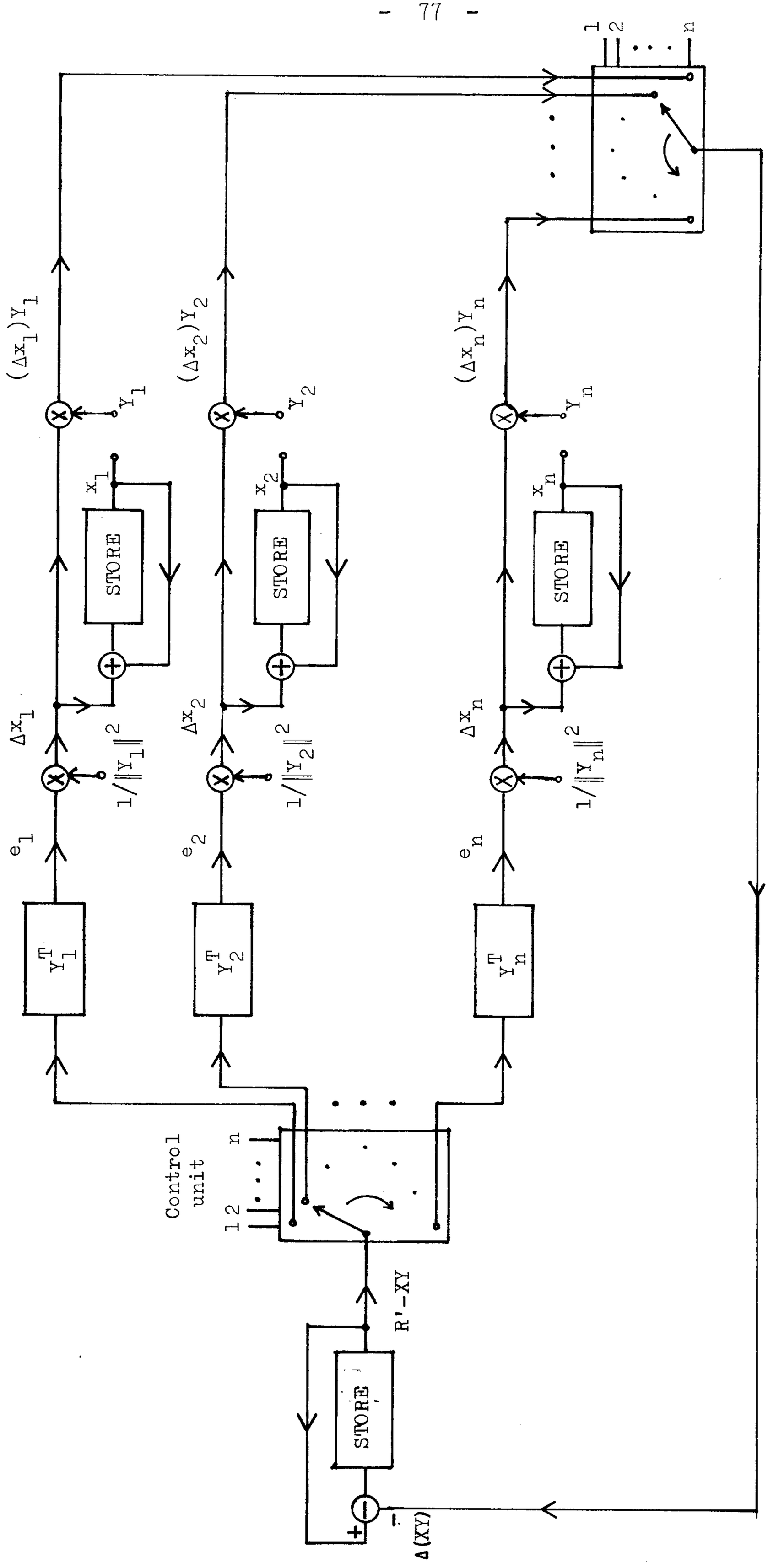


Fig. 3.3 Linear estimation process implemented as the Gauss-Seidel iterative process.

3.5 System 1

In the detection of the data symbol s_1 from R' (eqns. 3.5 - 3.9), system 1 uses an iterative process that is a simple development of the Gauss-Seidel iterative process described in section 3.4. The basic block diagram of the iterative process here is the same as that shown in Fig. 3.2, but now a constraint is placed on the vector X (eqn. 3.13). This constraint is such that any of the n components $x_1, x_2, \cdots x_n$ of X is only allowed to have a value of 1, -1, or 0 at all times during the iterative process. For convenience, let R_a be the n-component vector

$$R_a = R' - XY \tag{3.22}$$

where R' and Y are as defined by eqns. 3.7 and 3.9 respectively. R_a is therefore the vector at the input to the network Y^T shown in Fig. 3.2. The n-component vector E at the output of the network Y^T now becomes

$$E = (R' - XY)Y^T = R_a Y^T \tag{3.23}$$

and from eqn. 3.18, the hth component of E is now

$$e_h = (R' - XY)Y_h^T = R_a Y_h^T \tag{3.24}$$

for $h = 1,2,\cdots,n$. A more detailed diagram of the iterative process here is shown in Fig. 3.4. Various arrangements of this iterative process are investigated in systems 1.1 - 1.4, and are described as below.

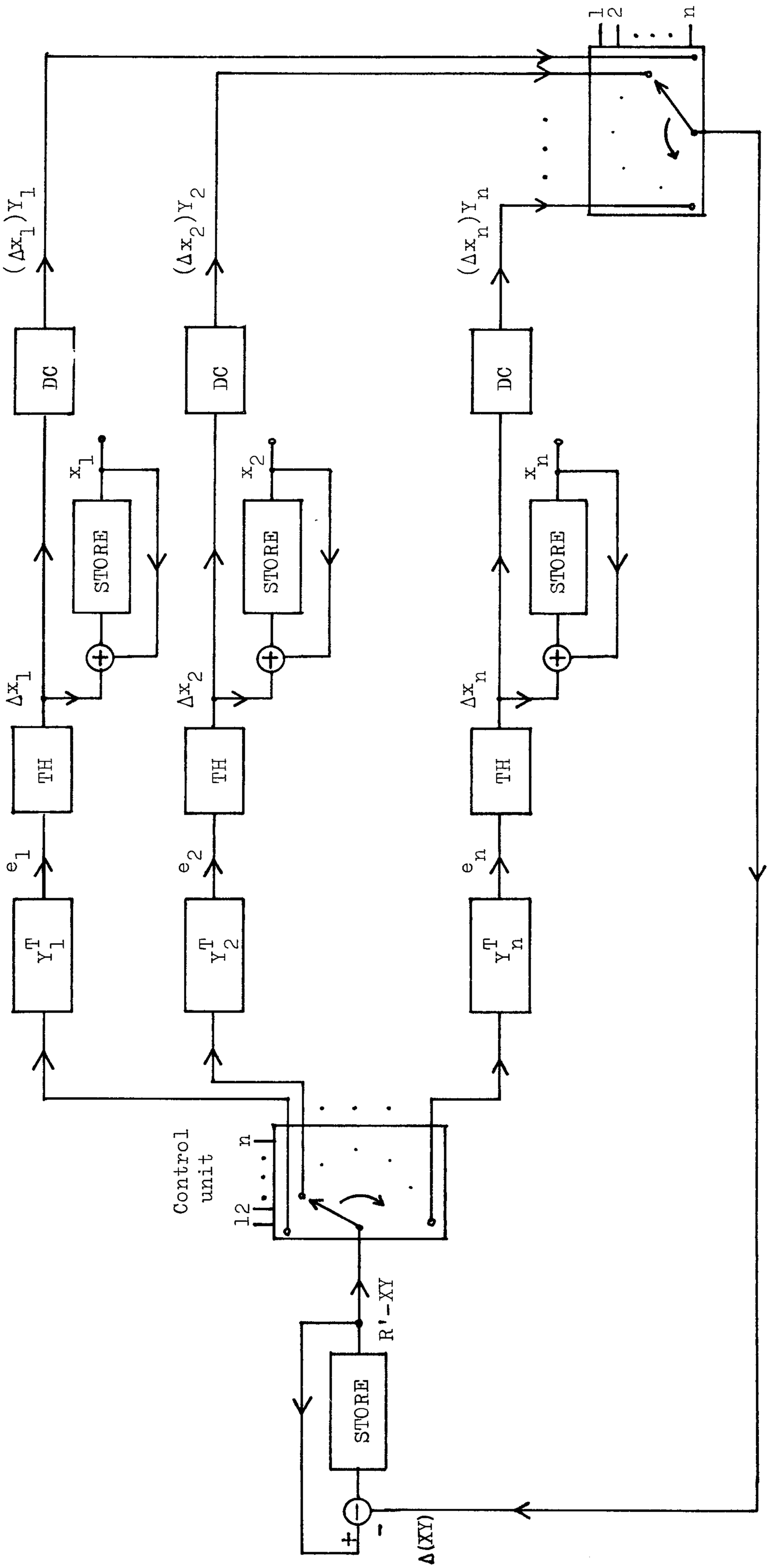


Fig. 3.4 A development of the Gauss-Seidel iterative process (Fig. 3.3).

TH : threshold detector

DC : decision device

System 1.1

The n-component vector X (eqn. 3.13) in Fig. 3.4 is initially set to zero. Thus, if X_o is the initial vector of X, then

$$X_o = \left[\overset{\longleftarrow n \longrightarrow}{0 \quad 0 \quad .. \quad 0} \right] \tag{3.25}$$

The initial vector of R_a (eqn. 3.22) in Fig. 3.4 is then set to $R' - X_o Y$. Having set the n components of X and R_a to their respective initial values, the iterative process of system 1.1 now operates as follows. The value of e_1 is first evaluated at the output of the network Y_1^T shown in Fig. 3.4 by using eqn. 3.24. This value of e_1 is then compared with its associated thresholds $\pm t_1$ to give the value of Δx_1, where t_1 has a positive value and will shortly be discussed. The value of Δx_1 is here the increment (adjustment) involved in the value of x_1 so that the new or updated value of x_1 is now given by

$$(x_1)_{new} = x_1 + \Delta x_1 \tag{3.26}$$

Thus, if $e_1 \geq t_1$, the value of Δx_1 is set to 1. The value of x_1 is then updated by using eqn. 3.26, and is here equal to 1. The vector R_a is then updated by removing from it all components of $(\Delta x_1 Y_1)$, so that the new vector is now given by

$$(R_a)_{new} = R_a - Y_1 \tag{3.27}$$

If $e_1 \leq -t_1$, Δx_1 and hence the new x_1 (eqn. 3.26) are both set to -1, and

$$(R_a)_{new} = R_a + Y_1 \tag{3.28}$$

If, however, $-t_1 > e_1 > t_1$, then Δx_1 is set to zero so that x_1 remains unchanged at zero, and

$$(R_a)_{new} = R_a \tag{3.29}$$

Eqns. 3.26 - 3.29 can be generalized to become

$$(x_h)_{new} = x_h + \Delta x_h \tag{3.30}$$

$$(R_a)_{new} = R_a - (\Delta x_h) Y_h \tag{3.31}$$

for $h = 1,2,\cdots,n$ for the whole iterative process of system 1.1. Having updated x_1 and R_a, the value of e_2 is next evaluated at the output of the network Y_2^T shown in Fig. 3.4. The value of Δx_2 is then determined by comparing e_2 with its associated thresholds $\pm t_2$ in the same way as that when the value of Δx_1 is determined from e_1 and $\pm t_1$. Thus, if $e_2 \geq t_2$, then Δx_2 is set to 1, and if $e_2 \leq -t_2$, then Δx_2 is set to -1. If, however, $-t_2 > e_2 > t_2$, then Δx_2 is set to zero. The value of x_2 and the vector R_a are then updated by using eqns. 3.30 and 3.31 respectively. The iterative process continues to operate in this way with the sequential adjustment (updating) of x_3 to x_n using the arrangement of Fig. 3.4. This completes the first iterative cycle of the iterative process and the whole cycle of operations just described is then repeated for the second iterative cycle starting with the evaluation of e_1 and so on.

The decision rule for determining the value of Δx_h, for any value of h between 1 to n, in the first iterative cycle of the iterative process here are summarised in Table 3.1.

In the second iterative cycle of the iterative process of system 1.1, the value of Δx_h is determined from the values of e_h and t_h according to Table 3.2. The operations involved here are otherwise the same as those of the first iterative cycle. Thus,

$$\Delta x_h = 1 \quad , \quad \text{if} \quad e_h \geq t_h$$
$$= 0 \quad , \quad \text{if} \quad t_h > e_h > -t_h$$
$$= -1 \quad , \quad \text{if} \quad -t_h \geq e_h$$

Table 3.1 Decision rule to determine the value of Δx_h (Fig. 3.4), for $h = 1,2,\cdot\cdot,n$, in the first iterative cycle of system 1.1.

$$\Delta x_h = 1 \quad , \quad \text{if} \quad x_h = 0 \quad \text{and} \quad e_h > 0$$
$$= -1 \quad , \quad \text{if} \quad x_h = 0 \quad \text{and} \quad e_h \leq 0$$
$$\Delta x_h = 2 \quad , \quad \text{if} \quad x_h = -1 \quad \text{and} \quad e_h > t_h$$
$$= 0 \quad , \quad \text{if} \quad x_h = -1 \quad \text{and} \quad e_h \leq t_h$$
$$\Delta x_h = -2 \quad , \quad \text{if} \quad x_h = 1 \quad \text{and} \quad e_h < -t_h$$
$$= 0 \quad , \quad \text{if} \quad x_h = 1 \quad \text{and} \quad e_h \geq -t_h$$

Table 3.2 Decision rule to determine the value of Δx_h (Fig. 3.4), for $h = 1,2,\cdot\cdot,n$, in the second and subsequent iterative cycles of system 1.1.

in the second iterative cycle here, the value of e_1 is first evaluated, at the output of the network Y_1^T shown in Fig. 3.4, as $R_a Y_1^T$ where R_a is here the vector of $R' - XY$ obtained at the end of the first iterative cycle. The value of Δx_1 is then determined from the values of e_1 and t_1 according to Table 3.2. This is followed by updating the value of x_1 and the vector R_a using eqns. 3.30 and 3.31 respectively. This completes the updating process for x_1 in the second iterative cycle, and the iterative process continues in this way with the sequential updating process for x_2 to x_n. It can be seen from Table 3.2 and eqn. 3.30 that, at the end of the second iterative cycle, all the $\{x_h\}$ for $h = 1,2,\cdots,n$ have been set to any of the two values 1 and -1 which are the two possible values of the data symbols $\{s_h\}$. Each subsequent iterative cycle operates in exactly the same way as for the second iterative cycle.

In the iterative process of system 1.1 just described, a counter is used to count the number of iterative cycles. When the counter exceeds a given threshold n_c, the iterative process is terminated and the value of x_1 (being equal to 1 or -1) is then taken as the detected value s_1' of s_1.

The number of multiplication and addition processes involved in the updating process of x_h $(1 \leq h \leq n)$ in the iterative process here and those in the Gauss-Seidel iterative process (section 3.4) are shown in Table 3.3. It should be noticed that although the value of e_h is evaluated as the product of the n-component row matrix R_a and the n-component column matrix Y_h^T (eqn. 3.24), only n-h+1 multiplications and n-h+1 additions are needed here because there are at most n-h+1 non-zero components in Y_h (eqn. 3.9). It should also be noticed that in the iterative process of system 1.1, the vector R_a can only be changed by 0, $\pm Y_h$, or $\pm 2Y_h$

since the value of Δx_h here can only have any of the values 0, ± 1, and ± 2 (Tables 3.1 and 3.2, and eqn. 3.31). Furthermore, multiplying a scalar quantity, say y_j, by 2 merely involves the moving of the binary coded numbers which form the quantity y_j, one place to the left.(C36) That is,

if y_j = 0 0 1 1 0 1 0 1 1 1 1 (11 bits)

then $2y_j$ = 0 0 1 1 0 1 0 1 1 1 1 0 (12 bits)

starting with the most significant bit.(C36) Consequently, no multiplication process is required in the updating process for R_a here (eqn. 3.31) as is shown in Table 3.3. Thus, as Table 3.3 suggests, the iterative process of system 1.1 requires fewer multiplication processes in the updating process for x_h in relation to those required in the Gauss-Seidel iterative process. The iterative process of system 1.1 therefore appears to be able to carry out the updating process more simply and rapidly.

	Iterative Process of System 1.1 (Fig. 3.4)	Gauss-Seidel Iterative Process (Fig. 3.3)
Evaluation of e_h (eqn. 3.24)	n-h+1 multiplications n-h+1 additions	n-h+1 multiplications n-h+1 additions
Determination of Δx_h	threshold decision (Table 3.1 or 3.2)	1 multiplication (eqn. 3.21)
Updating of x_h (eqn. 3.30)	1 addition	1 addition
Updating of R_a (eqn. 3.31)	n additions	n multiplications n additions

Table 3.3 Number of multiplications and additions involved in the updating process of x_h, for $h = 1,2,\cdots,n$.

System 1.2

System 1.2 is only slightly different from system 1.1 in that the iterative process here (Fig. 3.4) now begins with the updating process for x_n and works towards x_1 for each iterative cycle. This system is otherwise the same as system 1.1 which operates from x_1 to x_n for each iterative cycle.

System 1.3

System 1.3 differs from system 1.1 in that the iterative process here (Fig. 3.4) operates from x_1 to x_n and then back from x_n to x_1 for each iterative cycle. The amount of operations involved in each iterative cycle here is therefore twice of that for system 1.1. This system is otherwise the same as system 1.1.

System 1.4

This system is the same as system 1.3 except that the iterative process (Fig. 3.4) now operates from x_n to x_1 and then back from x_1 to x_n for each iterative cycle.

The threshold values $\{t_h\}$ used to determine the corresponding values of $\{\Delta x_h\}$ in Tables 3.1 and 3.2 for any of the systems 1.1 - 1.4, can be chosen to prevent the iterative process here from diverging and these values are determined in appendix A1. Divergence is here defined to have occurred when the Euclidean distance between the vector R' and the vector XY increases during the iterative process, bearing in mind that the vector X is here taken as the detected vector of S at the end of the iterative process and that R' is assumed to be given by eqn. 3.8. Thus, from appendix A1, the values of $\{t_h\}$ that prevent the iterative process here from diverging are

$$t_h \geq \|Y_h\|^2 \quad , \quad \text{if} \quad x_h = \pm 1$$

$$\text{and} \qquad t_h \geq \tfrac{1}{2}\|Y_h\|^2 \quad , \quad \text{if} \quad x_h = 0 \qquad (3.32)$$

for $h = 1,2,\cdots,n$, where the value of x_h shown here is its value before the value of Δx_h is determined from the value of t_h (Tables 3.1 and 3.2).

Four different versions of the values of $\{t_h\}$ to be used in each of the systems 1.1 - 1.4 have been considered and are shown in Table 3.4. It can be seen from eqn. 3.32 and Table 3.4 that, in the first iterative cycle, the iterative process of any of the systems 1.1 -1.4 with versions a and b is not prevented from diverging. It can also be seen that, in the second iterative cycle, the iterative process with any of the versions b, c, and d here is not prevented from diverging whenever the value of x_h is zero. The iterative process of any of the systems 1.1 - 1.4 with the various versions shown in Table 3.4 is otherwise prevented from diverging.

Versions	1st Cycle	2nd Cycle	All Subsequent Cycles
a	0	$\|Y_h\|^2$	$\|Y_h\|^2$
b	$\frac{1}{4}\|Y_h\|^2$	$\|Y_h\|^2$ if $x_h=\pm 1$ 0 if $x_h= 0$	$\|Y_h\|^2$
c	$\frac{1}{2}\|Y_h\|^2$	$\|Y_h\|^2$ if $x_h=\pm 1$ 0 if $x_h= 0$	$\|Y_h\|^2$
d	$\|Y_h\|^2$	$\|Y_h\|^2$ if $x_h=\pm 1$ 0 if $x_h= 0$	$\|Y_h\|^2$

Table 3.4 Four different versions of threshold values $\{t_h\}$, for $h = 1, 2, \cdots, n$, to be used to determine the values of $\{\Delta x_h\}$ (Table 3.1 or 3.2) in the iterative process of each of the systems 1.1 - 1.4.

Although the iterative process of any of the systems 1.1 - 1.4 with the various vesions shown in Table 3.4 is prevented from diverging in the third and subsequent iterative cycles, it does not necessarily converge in these iterative cycles since the value of Δx_h $(1 \leq h \leq n)$ can be zero (Table 3.2) so that the vector X and hence the Euclidean distance between the vector R' and the vector XY now remain unchanged. The meaning of convergence is the reverse of that defined for divergence previously. Computer simulation tests are therefore carried out for the various versions of systems 1.1 - 1.4 operating over channel A (Table 2.1)

to see if error-free detection in the absence of noise is achieved by any of these systems. In these computer simulation tests, 100 data symbols $\{s_i\}$ are transmitted in the absence of noise and the number of errors in detection is recorded for each value of n_c used. The value of n_c is the number of iterative cycles used in the iterative process here. An error in detection is defined to have occurred when the detected value s_1' of s_1 is not the same as the transmitted value of s_1. The results of the computer simulation tests are shown in Fig. 3.5. Clearly, as Fig. 3.5 shows, none of the various arrangements of systems 1.1 - 1.4 is able to achieve error-free detection over channel A (Table 2.1) in the absence of noise. These systems have therefore failed to operate correctly. Nevertheless, systems 1.1 - 1.4 are important in the sense that all the following systems of this chapter are developed from the arrangements used here.

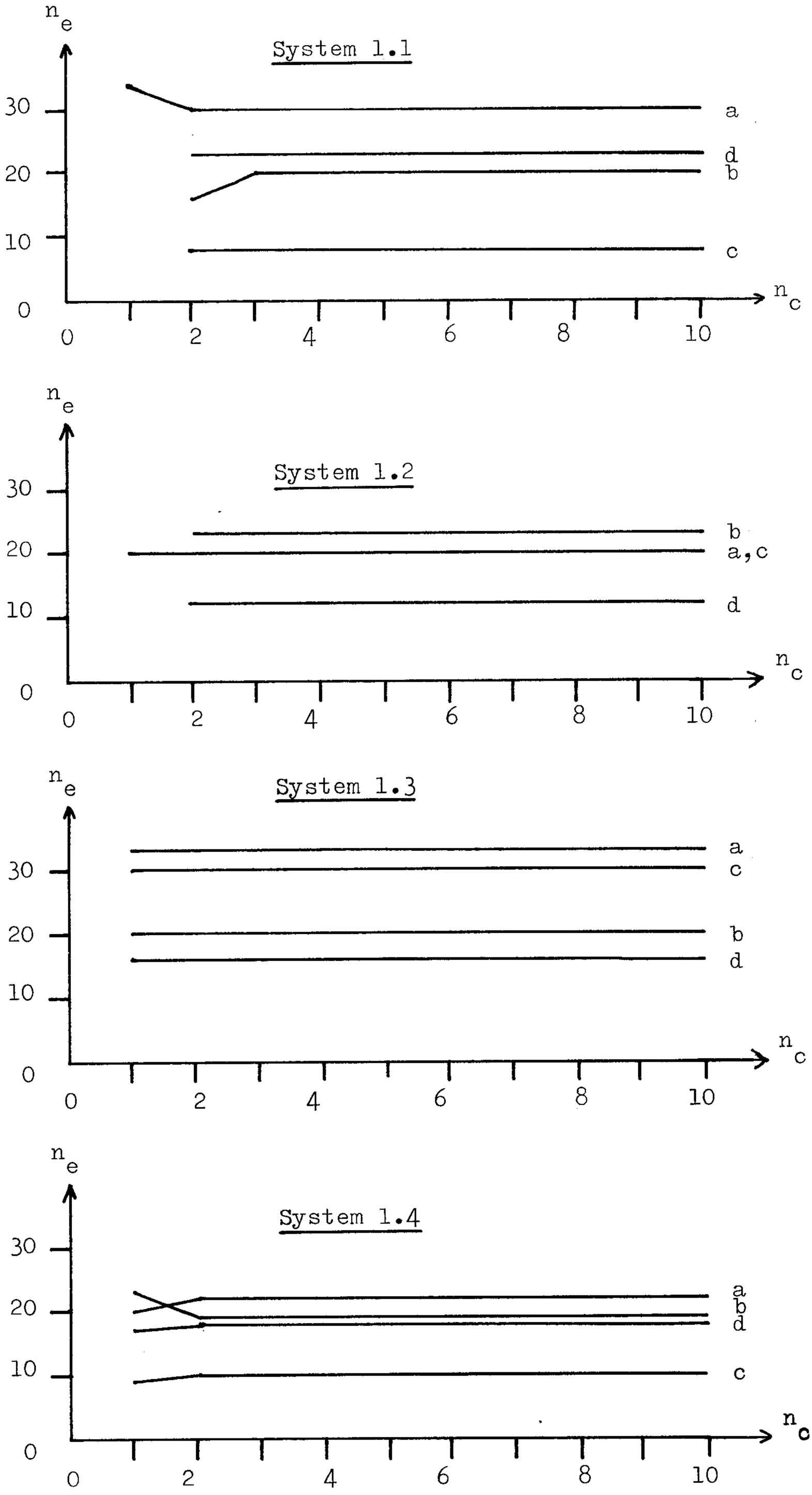


Fig. 3.5 Number of errors (n_e) in the detection of $\{s_i\}$ for systems 1.1 - 1.4 with versions a - d (Table 3.4) at various number of iterative cycles (n_c), when 100 data symbols $\{s_i\}$ are transmitted over channel A (Table 2.1) in the absence of noise.

3.6 System 2

System 2 is a modification of system 1, in that it uses two separate but similar iterative processes (instead of one in system 1) in the detection of the data symbol s_1 from R' (eqns. 3.5 - 3.9). Each of these two iterative processes employs the same piece of equipment shown in Fig. 3.4. The various arrangements of systems 1.1 - 1.4 are modified here to give the corresponding systems 2.1 - 2.4.

System 2.1

This is a modification of system 1.1. In the first iterative process here, the first component x_1 of the n-component vector X (eqn. 3.13) is set to 1 and held at this value during the whole of the iterative process. Thus, the iterative process operates by first setting the initial vector of X to

$$X_o = \left[1 \overbrace{\;0 \;\cdots\; 0\;}^{n-1} \right] \tag{3.33}$$

and the initial vector of R_a (eqn. 3.22) to

$$R' - X_o Y = R' - x_1 Y_1 = R' - Y_1 \tag{3.34}$$

where the $n \times n$ matrix Y and the n-component vector Y_1 are as defined by eqn. 3.9. The iterative process of system 1.1 is then applied here to the n-1 components $x_2, x_3, \cdots, x_n$ of X, starting from the updating process for x_2 to that for x_n for each iterative cycle. These operations are, of course, carried out by using the arrangement shown in Fig. 3.4. At the end of the iterative process when the number of iterative cycles exceeds the preset threshold value n_c, the Euclidean distance between R' and XY is measured as the quantity $\|R' - XY\|$. The whole process just described is then repeated for the second iterative

process but now x_1 is set and held at -1, so that the initial vector of X now becomes

$$X_o = \left[-1 \overbrace{\quad 0 \quad \cdots \quad 0 \quad}^{n-1} \right] \tag{3.35}$$

and the initial vector of R_a now becomes

$$R' - X_o Y = R' - x_1 Y_1 = R' + Y_1 \tag{3.36}$$

instead of those given by eqns. 3.33 and 3.34 respectively. The second iterative process is otherwise the same as the first iterative process and at the end of the iterative process here, the quantity $\|R' - XY\|$ is evaluated.

The detected value of s_1 is now taken to be the value of x_1 associated with the smaller of the two distances $\{\|R' - XY\|\}$.

System 2.2

System 2.2 is slightly different from system 2.1. The difference is that, while the iterative process of system 1.1 is applied in each of the two iterative processes of system 2.1 over the n-1 components $x_2, x_3, \cdots, x_n$, the iterative process of system 1.2 is applied here so that each of the two iterative processes of system 2.2 now begins with the updating process for x_n and works towards x_2 for each iterative cycle, this being carried out by using the arrangement shown in Fig. 3.4. System 2.2 is otherwise the same as system 2.1.

System 2.3

System 2.3 differs from system 2.1 in that, the iterative process of system 1.3 (instead of system 1.1) is used here so that each of the two iterative processes of system 2.3 now operates from x_2 to x_n and then back from x_n to x_2 for each iterative cycle, using the arrangement

shown in Fig. 3.4. This system is otherwise the same as system 2.1.

System 2.4

System 2.4 differs from system 2.3 in that, the iterative process of system 1.4 (instead of system 1.3) is used here so that each of the two iterative processes of system 2.4 now operates from x_n to x_2 and then back from x_2 to x_n for each iterative cycle. This system is otherwise the same as system 2.3.

Computer simulation tests are carried out for each of the systems 2.1 - 2.4 operating over channels A and C (Table 2.1) to see if error-free detection in the absence of noise is achieved by any of these systems. In these tests, 100 data symbols $\{s_i\}$ are transmitted in the absence of noise and the number of errors in detection is recorded for each value of n_c used, where n_c is the number of iterative cycles used in the given system. The results of the computer simulation tests are shown in Figs. 3.6 and 3.7, where the versions a, b, c, and d for each of the systems 2.1 - 2.4 are referred to the various versions of the threshold values $\{t_h\}$ (Table 3.4) to be used in the given system. Nevertheless, as Figs. 3.6 and 3.7 show, none of the arrangements of systems 2.1 - 2.4 is able to achieve error-free detection on both channels A and C even in the absence of noise. Systems 2.1 - 2.4 therefore do not operate correctly.

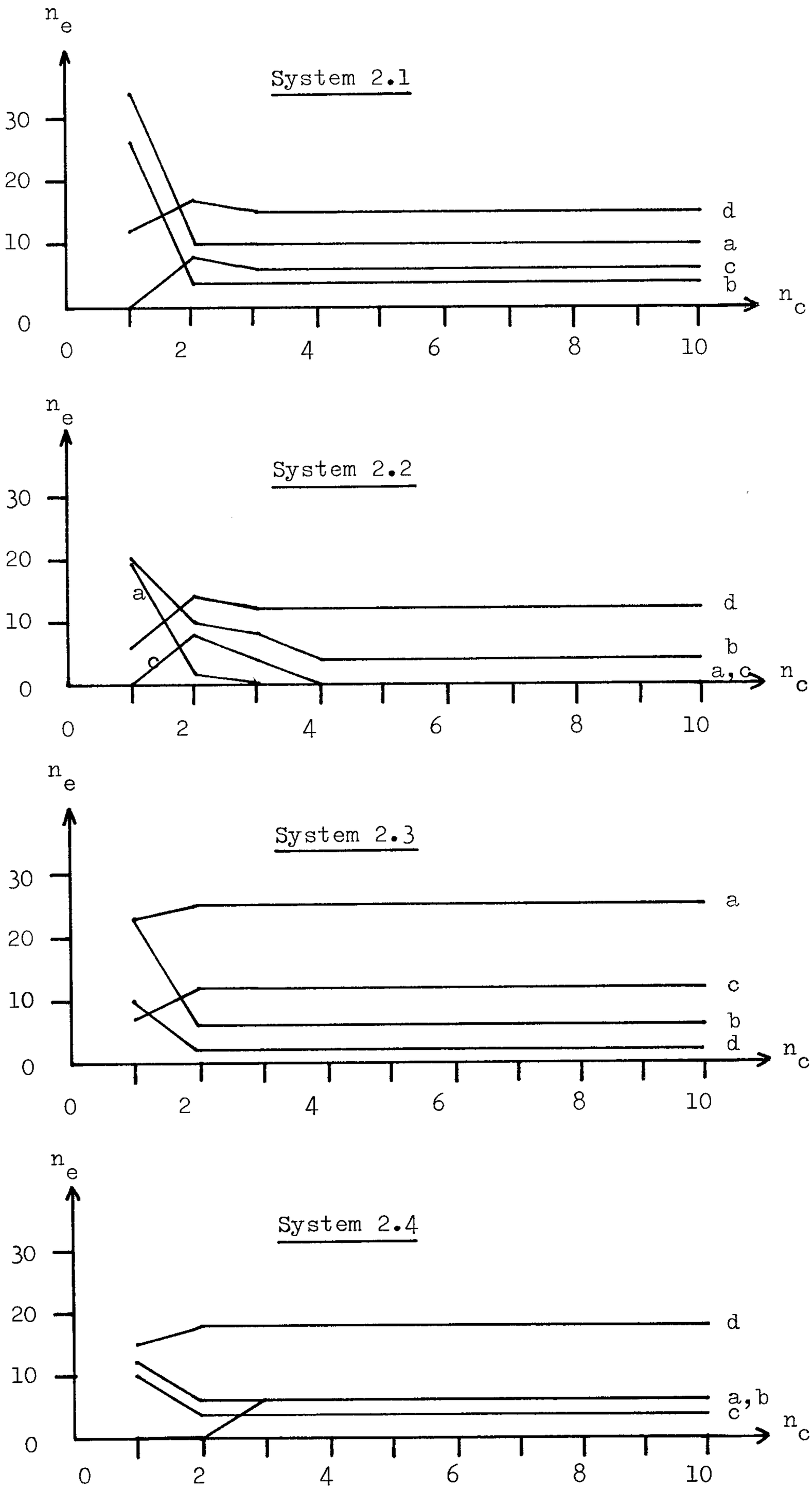


Fig. 3.6 Number of errors (n_e) in the detection of $\{s_i\}$ for systems 2.1 - 2.4 with versions a - d (Table 3.4) at various number of iterative cycles (n_c), when 100 data symbols $\{s_i\}$ are transmitted over channel A (Table 2.1) in the absence of noise.

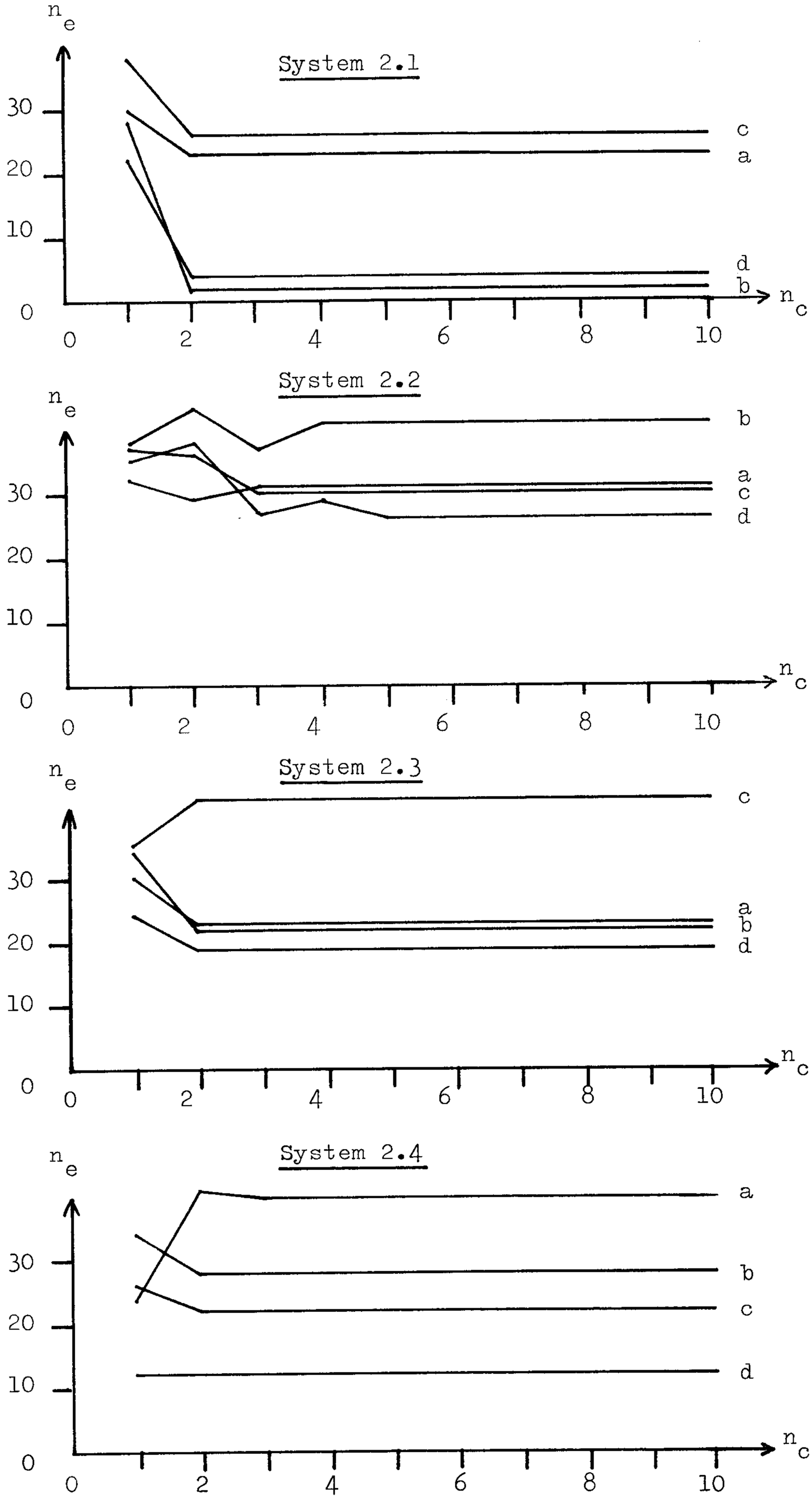


Fig. 3.7 Number of errors (n_e) in the detection of $\{s_i\}$ for systems 2.1 - 2.4 with versions a - d (Table 3.4) at various number of iterative cycles (n_c), when 100 data symbols $\{s_i\}$ are transmitted over channel C (Table 2.1) in the absence of noise.

3.7 System 3

Although the various arrangements of systems 1 and 2 do not give error-free detection in the absence of noise, the computer simulation results shown in Figs. 3.5 - 3.7 indicate that each of these systems appears to converge to a fixed error rate after a maximum of 4 iterative cycles, when operating over channel A or C (Table 2.1). It therefore appears that, with some slight modifications, these simple systems may be made to operate correctly with as few as 2 or 3 iterative cycles so that the advantage of high speed operation may be achieved here.

It is recalled that at the end of the second iterative cycle of the iterative process used in any of the systems 1 and 2, all the n components of X (eqn. 3.13) have been set to any of the two possible values of the data symbols $\{s_h\}$ so that X is from then on a possible vector of S (eqn. 3.5). The iterative process is then prevented from diverging (having a larger value of $\|R' - XY\|$) in the sequential updating process for the n components of X in the subsequent iterative cycles. This means that after the second iterative cycle, the iterative process here, effectively, proceeds to search for the possible vector of S associated with the minimum Euclidean distance $\|R' - SY\|$, subject to the constraint that only one component of S can be changed at a time. Thus, if X (being now a possible vector of S) has a smaller value of $\|R' - XY\|$ than those of the n possible vectors of S which differ from X in just one component, then no further changes can be made to any of the n components of this X which is of course not necessarily the possible vector of S associated with the minimum $\|R' - SY\|$. It therefore follows that the ability of the iterative process here to obtain the vector X as the possible vector of S associated with the smallest value of $\|R' - SY\|$, depends to some extent on the vector of X obtained at the end of the

second iterative cycle. Clearly, the vector of X obtained at the end of the second iterative cycle in the iterative process of any of the systems 1 and 2 is often very badly chosen so that the vector of X obtained at the end of the iterative process is very often not the possible vector of S associated with the smallest $\|R' - SY\|$ even in the absence of noise. It is for this reason that error-free detection in the absence of noise is not achieved by any of the systems 1 and 2 as is shown in Figs. 3.5 - 3.7. System 3 overcomes this problem by using a different initial vector X_o of X so that a better vector of X may be obtained when it first becomes a possible vector of S during the iterative process. The various arrangements of systems 1.1 - 1.4 are modified here to give the corresponding systems 3.1 - 3.4 described as below.

System 3.1

System 3.1 operates by using the n-1 components $s'_2, s'_3, \cdots, s'_n$ of the detected data-symbol vector S' obtained at the end of the previous detection process, as the initial values of $x_1, x_2, \cdots, x_{n-1}$ in the detection of the next data symbol. That is, the initial vector X_o of X is now set to

$$X_o = \begin{bmatrix} s'_2 & s'_3 & \cdots & s'_n & 0 \end{bmatrix} \qquad (3.37)$$

where $s'_2, s'_3, \cdots, s'_n$ are the detected values of the corresponding data symbols $s_2, s_3, \cdots, s_n$ (these being shifted one place to the front to become $s_1, s_2, \cdots, s_{n-1}$ respectively in the detection process now to be commenced) obtained at the end of the previous detection process. The initial vector of R_a (eqn. 3.22) is then set to $R' - X_o Y$. Having set the n components of X and R_a to their respective initial values, the iterative process of Fig. 3.4 is then applied here to the n components $x_1, x_2, \cdots, x_n$ of X. Thus, the value of e_1 is first

evaluated, at the output of the network Y_1^T shown in Fig. 3.4, as $R_a Y_1^T$ (eqn. 3.24). The value of Δx_1 is then determined from this value of e_1 according to Table 3.2. This is followed by updating the value of x_1 and the vector R_a using eqns. 3.30 and 3.31 respectively. This completes the updating process for x_1 in the first iterative cycle, and the iterative process continues in this way with the sequential updating process for x_2 to x_n, using the arrangement of Fig. 3.4. The whole process just described is then repeated, starting from the updating process for x_1 to that for x_n, for each of the subsequent iterative cycles. At the end of the iterative process when the number of iterative cycles exceeds the preset threshold value n_c, the vector X is taken as the detected data-symbol vector S' so that $s_1' = x_1$, $s_2' = x_2$, $\cdots$, $s_n' = x_n$. The value of s_1' is then accepted as the detected value of the data symbol s_1. The last n-1 components of S' are then shifted one place to the front to become the first n-1 components of the initial vector X_o of X for the next detection process.

It should be noticed that, in determining the values of $\{\Delta x_h\}$, Table 3.2 is used here for any of the iterative cycles. This is, of course, different from the arrangement of the iterative process used in system 1.1, where Table 3.1 is used in the first iterative cycle and Table 3.2 is used in each of the subsequent iterative cycles. Furthermore, the threshold values $\{t_h\}$ to be used in Table 3.2 for the iterative process of system 3.1 are also different from those given in Table 3.3, and are given as

$$\begin{aligned} t_h &= \|Y_h\|^2 \quad , \quad \text{if} \quad x_h = \pm 1 \\ &= 0 \quad , \quad \text{if} \quad x_h = 0 \end{aligned} \tag{3.38}$$

for $h = 1,2,\cdot\cdot,n$ where the value of x_h shown here is its value before the value of Δx_h is determined from the value of t_h (Table 3.2).

System 3.2

System 3.2 is slightly different from system 3.1. The difference is that the iterative process here now operates from the updating process for x_n to that for x_1 for each iterative cycle, using the the arrangement shown in Fig. 3.4. System 3.2 is otherwise the same as system 3.1.

System 3.3

System 3.3 differs from system 3.1 in that the iterative process here now operates from x_1 to x_n and then back from x_n to x_1 for each iterative cycle, using the arrangement of Fig. 3.4. The amount of operations involved in each iterative cycle here is therefore twice of that for system 3.1. This system is otherwise the same as system 3.1.

System 3.4

This system is the same as system 3.3 except that the iterative process here now operates from x_n to x_1 and then back from x_1 to x_n for each iterative cycle, using the arrangement of Fig. 3.4.

Systems 3.1 - 3.4 are in fact able to achieve error-free detection in the absence of noise and this is shown in appendix A2.

Computer simulation tests have been carried out to determine the tolerances to Gaussian noise of systems 3.1 - 3.4 operating over channel A (Table 2.1). The results of these tests are shown in Fig. 3.8, where the bit error rate (being the same as the symbol error rate here) is defined as

$$P_e = \frac{\text{Total number of errors in the detection of } \{s_i\}}{\text{Total number of data symbols } \{s_i\} \text{ transmitted}} \qquad (3.39)$$

The signal to noise ratio used in Fig. 3.8 is defined as

$$\text{SNR} = 10 \log_{10} \left(\frac{E'}{\frac{1}{2}N_o} \right) \quad \text{dB} \qquad (3.40)$$

where E' is the average transmitted energy per data symbol s_i, at the input to the transmission path (Fig. 2.1), and $\frac{1}{2}N_o$ is the two-sided power spectral density of the additive white Gaussian noise added at the input to the receiver filter. The 95% confidence limits of the results shown in Fig. 3.8 are about ± 0.5 dB. The optimum system shown here is the system described in section 3.3.

Clearly, none of the systems 3.1 - 3.4 has a performance anywhere near to that of the optimum system, and thus further modifications to these systems are needed. As both systems 3.2 and 3.4 seem to have better tolerances to noise than those of systems 3.1 and 3.3, it appears that the unknown component x_n of **X** should perhaps be determined first before the iterative process begins. This is carried out in system 3.5.

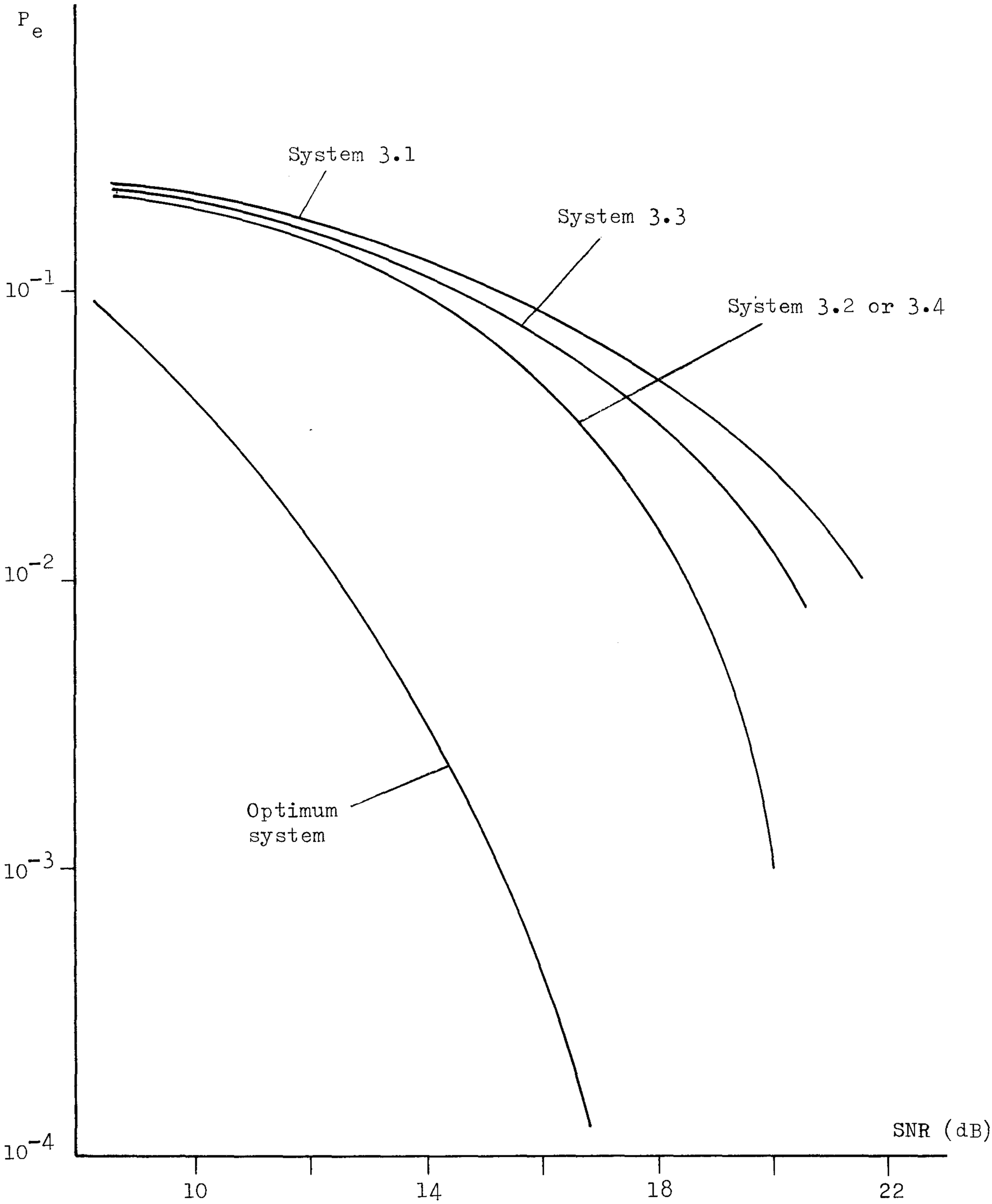


Fig. 3.8 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for systems 3.1 - 3.4 operating over channel A (Table 2.1).
Number of iterative cycles $n_c = 2$.

System 3.5

Thus, in any of the systems 3.1 - 3.4, the iterative process operates on the n-component vector R' (eqns. 3.4 and 3.7) to give the detected data-symbol vector S' whose last n-1 components s'_2, s'_3, ·· , s'_n are then used as the initial values of x_1, x_2, ·· , x_{n-1} for the next detection process. It can be seen from eqn. 3.4 that, only the first received component of s_{i+n-1}, the first two received components of s_{i+n-2}, the first three received components of s_{i+n-3}, and so on, are included in the n components of R', bearing in mind that s_i, ·· , s_{i+n-2}, s_{i+n-1} are here renamed as the n components s_1, ·· , s_{n-1}, s_n of S (eqn. 3.5) respectively. This implies that in the detection of S (as S') from R', errors are probably more likely to occur in the last few components of S' than to occur in the first few components of S' especially when the first few received components of the data symbol (or the first few components of the channel sampled impulse response) are of small magnitudes. Furthermore, if a component, say s'_h, of S' is in error (that is, $s'_h \neq s_h$), then by using it as the initial value of x_{h+1} for the next detection process doubles instead of removes the intersymbol interference of the data symbol s_{h+1} in the next detection process. It therefore appears, intuitively, that the tolerance to noise of the iterative process may sometimes be made higher by ignoring the detected values of the last few components of S so that only some of the last n-1 components of S' are now used as the initial values of the corresponding components of X for the next detection process. This is the basis of system 3.5.

Thus, in system 3.5, the initial vector X_o of **X** is set to

$$X_o = \left[s'_2 \quad s'_3 \quad \cdots \quad s'_{n-f+1} \overbrace{0 \quad \cdots \quad 0}^{f} \right] \qquad (3.41)$$

where $s'_2, s'_3, \cdots, s'_{n-f+1}$ are the detected values of the corresponding data symbols obtained at the end of the previous detection process, and f is an appropriate integer to be discussed shortly. The initial vector of R_a (eqn. 3.22) is then set to $\mathbf{R'} - \mathbf{X}_o\mathbf{Y}$. Having set the n components of **X** and R_a to their respective initial values, the iterative process of Fig. 3.4 is then applied here to the n components of **X**. However, the first iterative cycle of the iterative process here now begins with the updating process for x_{n-f+1}. This is carried out by first evaluating the value of e_{n-f+1} at the output of the network Y^T_{n-f+1} shown in Fig. 3.4. The value of Δx_{n-f+1} is then determined from this value of e_{n-f+1} according to Table 3.2 where the threshold values $\{t_h\}$ are as given in eqn. 3.38. This is then followed by updating the value of x_{n-f+1} and the vector R_a using eqns. 3.30 and 3.31 respectively. The process just described is the updating process for x_{n-f+1} and the iterative process continues in this way with the sequential updating process for x_{n-f+2} to x_n, using the arrangement of Fig. 3.4. The whole process just described is then repeated, starting from the updating process for x_1 to that for x_n, for each of the subsequent iterative cycles. The iterative process here is terminated when the number of iterative cycles exceeds the preset threshold value n_c. The vector **X** is then taken as the detected data-symbol vector S' so that $s'_1 = x_1, s'_2 = x_2, \cdots, s'_n = x_n$. However, only the first component s'_1 of S' is accepted as the detected value of the data symbol s_1. The components $s'_2, s'_3, \cdots, s'_{n-f+1}$ of S' are then stored and used as the initial values of $x_1, x_2, \cdots, x_{n-f}$ respectively for the next detection process.

It is interesting to note that, when f = 1, the initial vector X_o, of X, used in system 3.5 (eqn. 3.41) becomes identical to that (eqn. 3.37) used in any of the systems 3.1 - 3.4. Furthermore, when f = n, system 3.5 reduces exactly to system 1.1 which does not achieve error-free detection even in the absence of noise. The value of f to be used here should therefore be selected with care to ensure that error-free detection in the absence of noise is achieved by the system. It is shown in appendix A3 that, the condition for system 3.5 to achieve error-free detection in the absence of noise is that

$$d_h > 0 \qquad (3.42)$$

for $h = n-f+1, n-f+2, \cdots, n-1$ where

$$d_h = \frac{\|Y_h\|^2}{\|Y_h\|} - \sum_{j=1}^{n-h} \frac{|Y_{h+j} Y_h^T|}{\|Y_h\|} \qquad (3.43)$$

and Y_h is the n-component vector defined by eqn. 3.9. The value of f to be used in system 3.5 should therefore be such that the inequality of (3.42) is satisfied. The significance of the quantity d_h in eqn. 3.43 can be seen as follows. It can be seen from eqns. 3.8 and 3.24 that,

$$\begin{aligned} e_h &= (SY - XY)Y_h^T + WY_h^T \\ &= (s_1 - x_1)Y_1 Y_h^T + (s_2 - x_2)Y_2 Y_h^T + \cdots + (s_h - x_h)\|Y_h\|^2 \\ &\quad + \cdots + (s_n - x_n)Y_n Y_h^T + WY_h^T \end{aligned} \qquad (3.44)$$

where the noise component WY_h^T is a Gaussian random variable with zero mean and variance $\sigma^2 \|Y_h\|^2$. In the first iterative cycle of

the iterative process of system 3.5, the value of e_h is

$$e_h = (s_1 - x_1)Y_1Y_h^T + (s_2 - x_2)Y_2Y_h^T + \cdots + (s_{h-1} - x_{h-1})Y_{h-1}Y_h^T + s_h\|Y_h\|^2 + \cdots + s_nY_nY_h^T + WY_h^T \qquad (3.45)$$

for $h = n-f+1, n-f+2, \cdots, n$ where the values of $x_h, x_{h+1}, \cdots, x_n$ have not yet been determined (eqn. 3.41), and the values of $x_1, x_2, \cdots, x_{h-f}$ are the detected values of the corresponding data symbols obtained at the end of the previous detection process (eqn. 3.41). Eqn. 3.45 can be normalised by dividing each term by $\|Y_h\|$ so that the normalised noise component here now has a variance of σ^2 which is independent of the value of h. It can now be seen from eqn. 3.43 and the normalised form of eqn. 3.45 that, d_h is the difference between the normalised magnitude of the signal component (that associated with s_h) and the normalised maximum magnitude of the intersymbol interference component (that associated with $s_{h+1}, s_{h+2}, \cdots, s_n$) in the normalised value of e_h in the first iterative cycle of the iterative process of system 3.5, assuming that $s_1 = x_1, s_2 = x_2, \cdots, s_{h-1} = x_{h-1}$. It therefore appears that, at high signal to noise ratios when the values of x_1 to x_{h-1} have been updated to be the same as the corresponding values of s_1 to s_{h-1}, the value of d_h now determines to some extent the probability of x_h being updated to have the same value as that of s_h. In this case, a larger value of d_h is likely to give a lower probability of error in updating the value of x_h as s_h.

In order to achieve the highest tolerance to noise, it seems that the initial value of the component x_h of **X** should be set to zero (eqn. 3.41) so long as the detected value of the corresponding data symbol s_h obtained at the end of the previous detection process is more likely to be in

error than that obtained in the first iterative cycle of the next iterative process in system 3.5. That is, the detected data-symbol value obtained at the end of the previous detection process should not be used as the initial value of the corresponding component of X so long as the corresponding data symbol can be determined with a lower probability of error in the first iterative cycle of the next iterative process. Furthermore, as previous results have shown, the probability of error of the detected data-symbol value obtained at the end of a detection process appears to be dependent to a large extent on the initial vector X_o, of X, used in the iterative process here (which employs the arrangement of Fig. 3.4). Consequently, it appears intuitively that, in order to achieve the highest tolerance to noise, the probability of x_{n-f+1} being determined as the correct value of the data symbol s_{n-f+1} in the first iterative cycle of the iterative process here should be as high as possible, bearing in mind that x_{n-f+1} is the first component of X to be determined in the first iterative cycle of the iterative process here. Thus, at high signal to noise ratios when the initial values of x_1 to x_{n-f} (being here given by the detected values of the corresponding data symbols obtained at the end of the previous detection process) are the same as the values of the corresponding data symbols s_1 to s_{n-f}, the highest tolerance to noise of system 3.5 may be achieved if the value of f is such that

$$d_{n-f+1} \geq d_j \qquad (3.46)$$

for $j = 1,2,\cdots,n$ and $j \neq n-f+1$ where the quantity d_h is as defined by eqn. 3.43, bearing in mind that a larger value of d_{n-f+1} is now likely to give a lower probability of error in the detection of s_{n-f+1} in the first iterative cycle of the iterative process here.

Thus, it appears that the value of f to be used in the initial vector X_o (eqn. 3.41) of X in system 3.5 should be selected not only to ensure that the inequality of (3.42) is satisfied (for error-free detection in the absence of noise to be achieved), but also to maximize the value of d_{n-f+1} (eqn. 3.43) so as to hopefully achieve the highest tolerance to noise at high signal to noise ratios. It can be seen from Table 3.5 that, the inequality of (3.42) is only satisfied when f = 1, 2, or 3 for system 3.5 operating over channel A (Table 2.1). Consequently, only these values of f are considered in the computer simulation tests to determine the tolerance to noise of system 3.5 operating over channel A (Table 2.1).

The more promising results of the computer simulation tests are shown in Fig. 3.9, where the results are plotted as bit error rate versus signal to noise ratio. The definitions of the bit error rate (P_e) and the signal to noise ratio (SNR) here are as defined by eqns. 3.39 and 3.40 respectively. The 95% confidence limits of the curves shown in Fig. 3.9 are about ± 0.5 dB. The optimum system here is the system described in section 3.3.

It can be seen from Table 3.5 and Fig. 3.9 that, system 3.5 with f = 2 has a larger value of d_{n-f+1} and a higher tolerance to noise at high signal to noise ratios where the error rates drop below 10^{-2}, as compared to those of the given system with f = 3. This agrees with the analysis described previously that at high signal to noise ratios, the highest tolerance to noise of system 3.5 may be achieved if the value of f is such that the value of d_{n-f+1} is the maximum (inequality (3.46)). At low signal to noise ratios, Fig. 3.9 shows that system 3.5 with f = 3 appears to have a better tolerance to noise than that of the system with f = 2. One possible (but crude) reason to this is that, at low signal to noise ratios, the noise components dominate over the intersymbol

interference components in $\{e_h\}$ (eqn. 3.44) so that the intersymbol interference components now have a smaller effect on the probability of error in the detection of the data symbol s_{n-f+1} from e_{n-f+1} in the first iterative cycle of the iterative process here. It therefore appears intuitively that the system that has a larger value of $\|Y_{n-f+1}\|$ (which is the magnitude of the signal component in e_{n-f+1} associated with s_{n-f+1}) instead of d_{n-f+1} is now likely to have a higher tolerance to noise. It can be seen from Table 3.5 that system 3.5 with f = 3 has a larger value of $\|Y_{n-f+1}\|$ than that of the system with f = 2, and thus the tolerance to noise of the system with f = 3 may be expected to be higher than that of the system with f = 2 at low signal to noise ratios as is shown in Fig. 3.9.

It can be from the simulation results shown in Figs. 3.8 and 3.9 that, system 3.5 has the best tolerance to noise of all the systems developed and studied in this section, when operating over channel A (Table 2.1). In particular, the tolerance to noise of system 3.5 with f = 2 and $n_c = 2$ is at least 1.5 dB better than that of any of the systems 3.1 - 3.4, at an error rate of 10^{-3}. The quantity n_c is the number of iterative cycles used in the system. If the operations involved in the updating process for one component of the vector X is referred to as one sequential operation, then it can be seen from the descriptions of the operations of the various systems here that the number of sequential operations n_s involved in the iterative process of system 3.1 or 3.2 is

$$n_s = (n)(n_c) \tag{3.47}$$

and that involved in the iterative process of system 3.3 or 3.4 is

$$n_s = 2(n)(n_c) \tag{3.48}$$

where n is the number of sample values $\{r'_h\}$ (eqn. 3.2) used in the

iterative process here. The number of sequential operations involved in system 3.5 is, however, given by

$$n_s = (n)(n_c - 1) + f \qquad (3.49)$$

The value of n_s involved in each of the systems 3.1 - 3.5 shown in Figs. 3.8 and 3.9 is shown in Table 3.6. As Table 3.6 shows, the number of sequential operations involved in system 3.5 here appears to be smaller than that involved in any of the systems 3.1 - 3.4.

System 3.5 is therefore the most promising of the systems developed and studied in this section.

f (eqn. 3.41)	h (=n-f+1)	$\frac{\|Y_h\|^2}{\|Y_h\|}$ or $\|Y_h\|$	$\frac{\|Y_{h+j}Y_h^T\|}{\|Y_h\|}$	d_h (eqn. 3.43)
1	8	0.167	0.000	+0.167
2	7	0.500	0.157	+0.343
3	6	0.866	0.612	+0.254
4	5	0.986	1.178	-0.192
5	4	1.000	1.466	-0.466
6	3	1.000	1.466	-0.466
7	2	1.000	1.466	-0.466
8	1	1.000	1.466	-0.466

Table 3.5 Values of $\{d_h\}$ for channel A (Table 2.1). $n = 8$.
The sampled impulse response of channel A is
0.167 0.471 0.707 0.471 0.167

Systems	Number of Iterative Cycles n_c	Number of Sequential Operations n_s
3.1	2	16
3.2	2	16
3.3	2	32
3.4	2	16
3.5 (f=2)	1	2
3.5 (f=2)	2	10
3.5 (f=3)	2	11

Table 3.6 Number of iterative cycles n_c and number of sequential operations n_s involved in systems 3.1 - 3.5 (Figs. 3.8 and 3.9)

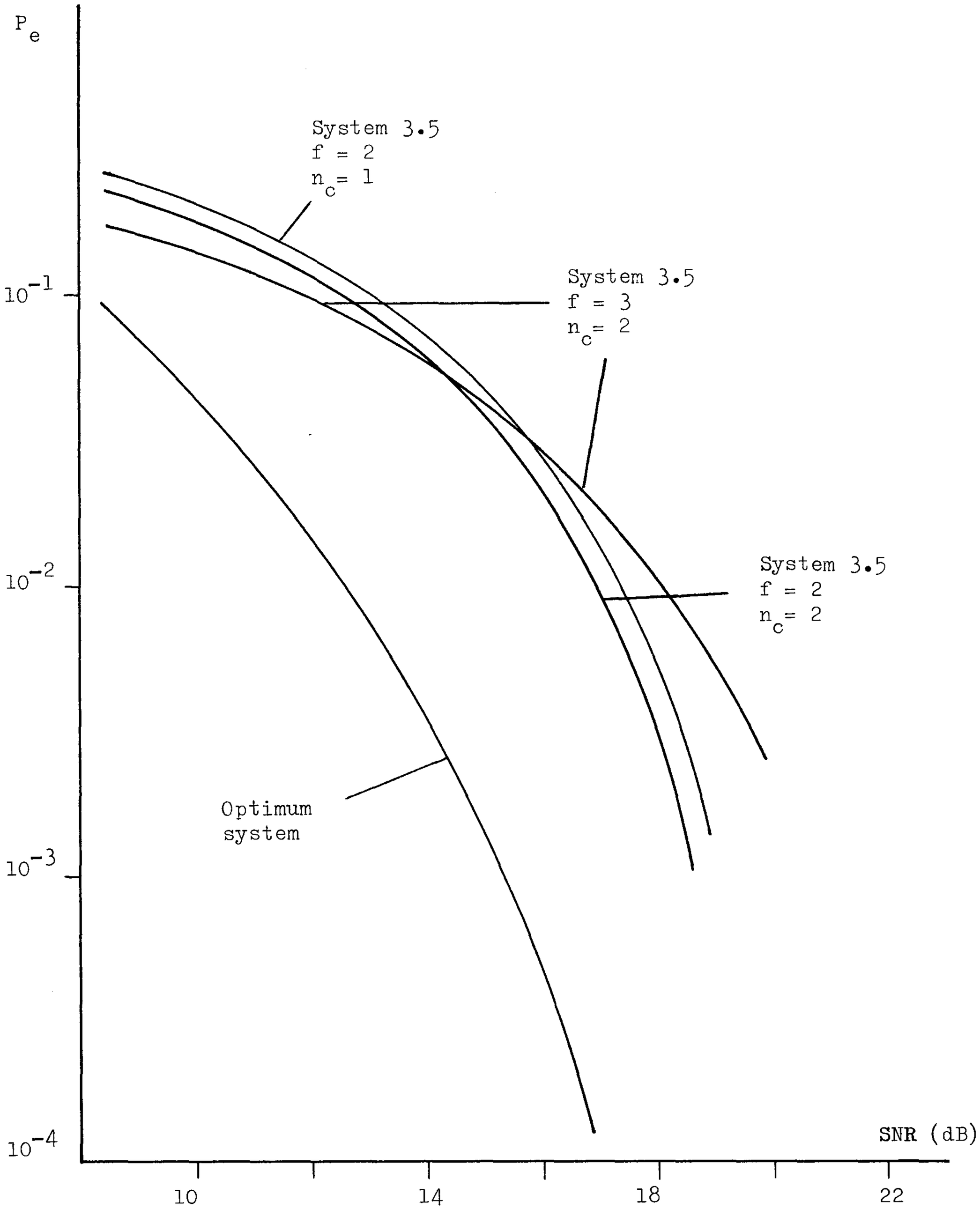


Fig. 3.9 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 3.5 operating over channel A (Table 2.1).

n_c : number of iterative cycles.

f : integer defined in eqn. 3.41.

3.8 System 4

System 4 is a modification of system 3. The modification here is the same as the modification involved in converting system 1 to system 2. Thus, while system 3 uses only one iterative process, system 4 uses two separate but similar iterative processes in the detection of the data symbol s_1. Each of these two iterative processes of system 4 employs the same piece of equipment shown in Fig. 3.4. The various arrangements of systems 3.1 - 3.5 are modified here to give the corresponding systems 4.1 - 4.5.

System 4.1

This is a modification of system 3.1. In the first iterative process here, the first component x_1 of the n-component vector X (eqn. 3.13) is set to 1 and held at this value during the whole of the iterative process. The iterative process here now begins to operate by setting the initial vector of X to

$$X_o = \begin{bmatrix} 1 & s'_3 & s'_4 & \cdots & s'_n & 0 \end{bmatrix} \qquad (3.50)$$

where $s'_3, s'_4, \cdots, s'_n$ are the last n-2 components of the detected data-symbol vector S' (of S) obtained at the end of the previous detection process. The initial vector of R_a (eqn. 3.22) is then set to $R' - X_o Y$, where R' is the n-component vector defined by eqn. 3.7 and Y is the $n \times n$ matrix defined by eqn. 3.9. The iterative process of system 3.1 is then applied here to the n-1 components $x_2, x_3, \cdots, x_n$ of X, starting from the updating process (Table 3.3) for x_2 to that for x_n for each iterative cycle, using the arrangement shown in Fig. 3.4. At the end of the iterative process when the number of iterative cycles exceeds the preset threshold value n_c, the quantity $\|R' - XY\|$ is

measured and stored together with the corresponding n components of X. The whole process just described is then repeated for the second iterative process but now x_1 is set and held at -1, so that the initial vector of X now becomes

$$X_o = \begin{bmatrix} -1 & s'_3 & s'_4 & \cdots & s'_n & 0 \end{bmatrix} \qquad (3.51)$$

which is the same as that (eqn. 3.50) used in the first iterative process except for the first component. The initial vector of R_a for the second iterative process is, of course, the vector $R' - X_o Y$ where X_o is as given by eqn. 3.51. The second iterative process is otherwise the same as the first iterative process, and at the end of this second iterative process, the quantity $\|R' - XY\|$ is measured and stored together with the corresponding vector X.

The detected vector S' of S is now taken to be the vector of X associated with the smaller of the two distances $\{\|R' - XY\|\}$. However, only the first component s'_1 of S' is accepted as the detected value of the data symbol s_1. The last n-2 components $s'_3, s'_4, \cdots, s'_n$ of S' are then shifted one place to the front and used as the initial values of the corresponding n-2 components $x_2, x_3, \cdots, x_{n-1}$ of X for the next detection process.

System 4.2

System 4.2 differs from system 4.1 in that, the iterative process of system 3.2 (instead of system 3.1) is used here so that each of the two iterative processes of system 4.2 now operates from the updating process for x_n and works towards x_2 for each iterative cycle, using the arrangement shown in Fig. 3.4. This system is otherwise the same as system 4.1.

System 4.3

System 4.3 differs from system 4.1 in that, the iterative process of system 3.3 is now used here so that each of the two iterative processes of system 4.3 now operates from x_2 to x_n and then back from x_n to x_2 for each iterative cycle, using the arrangement shown in Fig. 3.4. This system is otherwise the same as system 4.1.

System 4.4

System 4.4 is the same as system 4.3 except that each of the two iterative processes here now operates from x_n to x_2 and then back from x_2 to x_n for each iterative cycle.

System 4.5

This is a modification of system 3.5. The modification here is that two (instead of one in system 3.5) separate but similar iterative processes are now used in the detection of the data symbol s_1 from R'. In the first iterative process here, x_1 is set to 1 and held at this value during the whole of the iterative process. The initial vector X_o of X is now set to

$$X_o = \left[1 \quad s'_3 \quad s'_4 \quad \cdots \quad s'_{n-f+1} \overbrace{0 \quad \cdots \quad 0}^{f} \right] \qquad (3.52)$$

where $s'_3, s'_4, \cdots, s'_{n-f+1}$ are the detected values of the corresponding data symbols obtained at the end of the previous detection process, and f is an appropriate integer between 1 to n inclusive. The initial vector of R_a (eqn. 3.22) is then set to $R' - X_o Y$, using the vector of X_o given by eqn. 3.52. The iterative process of system 3.5 is then applied here starting from the updating process for x_2 to that for x_n for each iterative cycle (except the first iterative cycle where the iterative process operates from x_{n-f+1} to x_n), using the arrangement

shown in Fig. 3.4. At the end of the iterative process when the number of iterative cycles exceeds the preset threshold value n_c, the quantity $\|R' - XY\|$ is measured and stored together with the corresponding vector of X. The whole process just described is then repeated for the second iterative process but now x_1 is set and held at -1, so that the initial vector of X now becomes

$$X_o = \left[-1 \quad s'_3 \quad s'_4 \quad \cdots \quad s'_{n-f+1} \overbrace{0 \quad \cdots \quad 0}^{f} \right] \qquad (3.53)$$

and the initial vector of R_a becomes $R' - X_o Y$ with this vector of X_o. The second iterative process is otherwise the same as the first iterative process, and at the end of this second iterative process, the quantity $\|R' - XY\|$ is measured and stored together with the corresponding vector X.

The detected vector S' of S is now taken to be the vector of X associated with smaller of the two quantities $\{\|R' - XY\|\}$. However, only the first component s'_1 of S' is accepted as the detected value of the data symbol s_1. The n-f-1 components $s'_3, s'_4, \cdots s'_{n-f+1}$ of S' are then shifted one place to the front and used as the initial values of the corresponding n-f-1 components $x_2, x_3, \cdots, x_{n-f}$ of X for the next detection process.

The conditions for systems 4.1 - 4.5 to achieve error-free detection in the absence of noise are now considered. It is recalled that, each of the systems 4.1 - 4.4 uses the corresponding iterative process of each of the systems. 3.1 - 3.4 to operate over the n-1 components $x_2, x_3, \cdots, x_n$ of X in each of its two iterative processes. The value of x_1 is 1 in one of the iterative processes and is -1 in the other. Clearly, one of these two values of x_1 must be the correct value which is the value of the data symbol s_1. It is shown in appendix A2 that, in the absence of noise, all the components of X obtained at the end of the

iterative process of any of the systems 3.1 - 3.4 are always equal to the corresponding components of the data-symbol vector S. It follows that, in the absence of noise, the n-1 components $x_2, x_3, \cdots, x_n$ of X obtained at the end of the iterative process associated with the correct value of x_1 in any of the systems 4.1 - 4.4 must also always be equal to the corresponding components of S, so that X = S and the associated distance $\|R' - XY\|$ is always equal to zero here, bearing in mind that R' = SY here. Consequently, this vector of X (where X = S) is always taken as the detected vector S' of S, and thus error-free detection must always be achieved by any of the systems 4.1 - 4.4 in the absence of noise.

Similarly, system 4.5 uses the iterative process of system 3.5 to operate over the n-1 components $x_2, x_3, \cdots, x_n$ of X in each of its two iterative processes having the two possible values of x_1. It is shown in appendix A3 that, if the inequality of (3.42) is satisfied, then in the absence of noise, the components of X obtained at the end of the iterative process of system 3.5 are always equal to the corresponding components of S. Clearly, if this inequality is satisfied, then in the absence of noise, the n-1 components $x_2, x_3, \cdots, x_n$ of X obtained at the end of the iterative process associated with the correct value of x_1 in system 4.5 must also always be equal to the corresponding components of S, so that X = S and the associated distance $\|R' - XY\|$ is always equal to zero here. The vector of X for which X = S is now always taken as the detected vector S' of S, and hence error-free detection is achieved here. Consequently, the condition for system 4.5 to achieve error-free detection in the absence of noise is the same as that for system 3.5 and is given as the inequality of (3.42).

It can be seen from eqns. 3.37, 3.41, and 3.50 - 3.53 that one of the two iterative processes of each of the systems 4.1 - 4.5 uses exactly the same initial vector of X as that of the iterative process of each of the respective systems 3.1 - 3.5. Furthermore, each iterative process of each of the systems 4.1 - 4.5 operates in the same way as the iterative process of each of the respective systems 3.1 - 3.5. Consequently, each of the systems 4.1 - 4.5 is likely to have the same or higher tolerance to noise in relation to that of each of the respective systems 3.1 - 3.5, since more iterative processes are being used in each of the systems 4.1 - 4.5 to search for the possible vector of S associated with the smallest value of the Euclidean distance $\|R' - SY\|$.

Computer simulation tests have been carried out to determine the tolerances to Gaussian noise of the systems 4.1 - 4.5 operating over channel A (Table 2.1), and the more promising results of these tests are shown in Fig. 3.10 where the results are plotted as bit error rate versus signal to noise ratio. The definitions of the bit error rate (P_e) and the signal to noise ratio (SNR) here are as given by eqns. 3.39 and 3.40 respectively. The 95% confidence limits of the results in Fig. 3.10 are about ± 0.5 dB. The performance of the optimum system described in section 3.3 is also included here so as to show the relative performances of the systems 4.1 - 4.5 to this optimum system. Table 3.7 also shows the number of sequential operations n_s involved in each of the systems 4.1 - 4.5 shown in Fig. 3.10, where a sequential operation is defined as the operations involved in updating one component of X in the iterative process of the given system. It can be seen from the descriptions of the operations of the systems 4.1 - 4.5 given previously that the number of sequential operations n_s involved in the detection process is

$$n_s = 2(n - 1)(n_c) \tag{3.54}$$

for each of the systems 4.1 and 4.2, and is

$$n_s = 4(n - 1)(n_c) \tag{3.55}$$

for each of the systems 4.3 and 4.4, and is

$$n_s = 2(n - 1)(n_c - 1) + 2(f) \tag{3.56}$$

for system 4.5, where n is the number of sample values $\{r'_h\}$ (eqn. 3.2) used in the iterative process here, n_c is the number of iterative cycles, and f is the integer defined in eqn. 3.52 or 3.53.

Systems	Number of Iterative Cycles n_c	Number of Sequential Operations n_s
4.1	2	28
4.2	2	28
4.3	2	56
4.4	2	28
4.5 (f=2)	2	18
4.5 (f=3)	2	20

Table 3.7 Number of iterative cycles n_c and number of sequential operations n_s involved in systems 4.1 - 4.5 (Fig. 3.10).

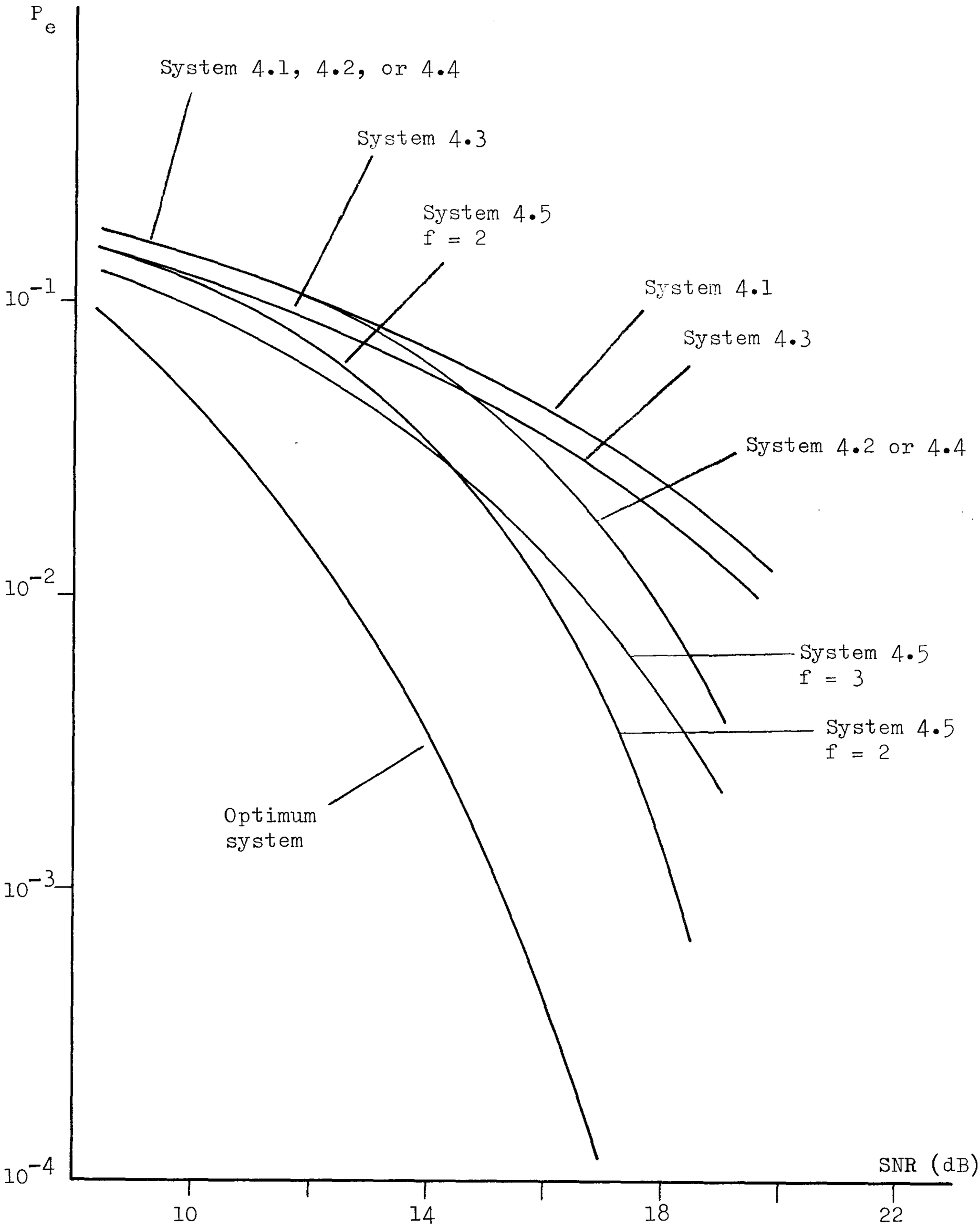


Fig. 3.10 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for systems 4.1 - 4.5 operating over channel A (Table 2.1).
Number of iterative cycles $n_c = 2$.
f : integer defined in eqns. 3.50 and 3.51.

It can be seen from Fig. 3.10 that, system 4.5 has the best tolerance to noise of all the systems studied in this section. Furthermore, Table 3.7 shows that, the number of sequential operations involved in the detection process of system 4.5 is also fewer than any of those of the systems 4.1 - 4.4. System 4.5 is therefore the most promising system of all the various arrangements of system 4 studied here.

It can be seen from Figs. 3.9 and 3.10 that, system 4.5 has a better tolerance to noise than that of system 3.5 at low signal to noise ratios. As the signal to noise ratio increases, the performance of system 3.5 approaches to that of system 4.5 indicating that only one of the two iterative processes used in system 4.5 is useful here. Modifications are therefore needed to overcome this weakness of system 4.5 and this leads to the development of system 5.

Since systems 3.5 and 4.5 have been shown to be the most promising systems of all the various arrangements of systems 3 and 4 respectively, system 3.5 will from now on be referred to as system 3 and system 4.5 will from now on be referred to as system 4. Consequently, only the arrangement of the iterative process used in system 3.5 or 4.5 will be considered in system 5. That is, the iterative process used in system 5 will start with the sequential operation associated with x_{n-f+1} and end with the sequential operation associated with x_n in the first iterative cycle, and then from x_1 to x_n in each subsequent iterative cycle.

3.9 System 5

It has previously been shown that the iterative process that employs the arrangement of Fig. 3.4 is such that, after the n-component vector X (eqn. 3.13) has been updated to become a possible vector of S, the iterative process then, effectively, proceeds to search for the possible vector of S associated with the minimum distance $\|R' - SY\|$, subject to the constraint that only one component of S can be changed at a time. It is this constraint that sometimes prevents the iterative process from reaching the possible vector of S associated with the minimum distance. As previous results have shown, the ability of this iterative process to obtain the vector X as the possible vector of S associated with the minimum distance appears to be dependent critically on the initial vector of X used in the iterative process, bearing in mind that the tolerance to noise of a system depends to a large extent on this ability of the iterative process used in the system at high signal to noise ratios. The system that uses an initial vector X_o (of X) for which the corresponding n-component vector X_oY is nearer to the n-component signal vector SY (eqn. 3.8) is here likely to have a higher tolerance to noise at high signal to noise ratios. It therefore appears that, the two initial vectors of $\{XY\}$ used in system 4 may be so far apart that at high signal to noise ratios one of them becomes very far away from the signal vector SY and is hence very unlikely to be considered useful in the detection process. This provides a possible explanation to the results shown in Figs. 3.9 and 3.10 where the performance of system 3 appears to approach that of system 4 at high signal to noise ratios. It can be seen from eqns. 3.52 and 3.53 that, the two initial vectors of $\{XY\}$ used in system 4 differ by $\pm 2Y_1$ and so the distance between them

is the Euclidean norm of $\pm 2Y_1$ and is given by

$$\| \pm 2Y_1 \| = 2 \left(y_0^2 + y_1^2 + \cdots + y_n^2 \right)^{\frac{1}{2}} \qquad (3.57)$$

where $y_h = 0$ for $h > g$. The n-component vector Y_1 here is as defined by eqn. 3.9, and $y_0, y_1, \cdots, y_g$ are the g+1 components of the channel sampled impulse response. System 5 is a simple development of system 4 in that the two initial vectors of $\{XY\}$ are here selected to be separated by a distance smaller than that given by eqn. 3.57. The detection process of system 5 is otherwise similar to that of system 4.

Thus, system 5 uses two separate but similar iterative processes in the detection of the data symbol s_1. Each of these two iterative processes employs the same piece of equipment shown in Fig. 3.4. The two initial vectors of $\{X\}$ to be used here are selected to be such that

$$(X_o)_1 = \left[s'_2 \;\; s'_3 \;\; \cdot\cdot \;\; s'_{n-f} \;\; s'_{n-f+1} \;\; \overbrace{0 \;\; \cdot\cdot \;\; 0}^{f} \right] \qquad (3.58)$$

$$(X_o)_2 = \left[s'_2 \;\; s'_3 \;\; \cdot\cdot \;\; s'_{n-f} \;\; -s'_{n-f+1} \;\; \overbrace{0 \;\; \cdot\cdot \;\; 0}^{f} \right] \qquad (3.59)$$

and $f < g$, where $s'_2, s'_3, \cdot\cdot, s'_{n-f+1}$ are the detected values of the corresponding data symbols obtained at the end of the previous detection process. Clearly, these two vectors differ only in the value of x_{n-f}, and the two corresponding initial vectors $\{X_o Y\}$, of $\{XY\}$, are here separated by a distance of

$$\| \pm 2Y_{n-f} \| = 2 \left(y_0^2 + y_1^2 + \cdots + y_f^2 \right)^{\frac{1}{2}} \qquad (3.60)$$

which is obviously smaller than that given by eqn. 3.57 (for system 4) since $f < g$ here. Thus, in the first iterative process of system 5, the initial vector of X is set to $(X_o)_1$ (eqn. 3.58) and the initial vector of R_a (eqn. 3.22) is set to $R' - (X_o)_1 Y$, where R' is the n-component vector defined by eqn. 3.7 and Y is the $n \times n$ matrix defined by eqn. 3.9.

The iterative process of system 3.5 is then applied here starting from the updating process for x_1 to that for x_n for each iterative cycle (except the first iterative cycle where the iterative process operates from x_{n-f+1} to x_n), using the arrangement shown in Fig. 3.4. The updating process for x_h $(1 \leq h \leq n)$ here includes the evaluation of the quantity e_h at the output of the network Y_h^T shown in Fig. 3.4, the determination of Δx_h using the threshold device which operates according to Table 3.2 and eqn. 3.38, and the updating of x_h and R_a using eqns. 3.30 and 3.31 respectively. The iterative process here is terminated when the number of iterative cycles exceeds the preset threshold value n_c. The distance $\|R' - XY\|$ is then measured and stored together with the corresponding vector of X. The whole process just described is then repeated for the second iterative process but now the initial vector of X is set to $(X_o)_2$ (eqn. 3.59) and the initial vector of R_a is set to $R' - (X_o)_2 Y$. The second iterative process is otherwise the same as the first iterative process, and at the end of this second iterative process, the quantity $\|R' - XY\|$ is measured and stored together with the corresponding vector X. The vector $R' - XY$ obtained at the end of an iterative process is, of course, the vector R_a obtained at the end of the iterative process. The detected data-symbol vector S' of S is now taken to be the vector of X associated with smaller of the two quantities $\{\|R' - XY\|\}$. However, only the first component s_1' of S' is accepted as the detected value of the data symbol s_1. The n-f components $s_2', s_3', \cdots s_{n-f+1}'$ of S' are then shifted one place to the front and used as the initial values of the corresponding n-f components $x_1, x_2, \cdots, x_{n-f}$ of X for the next detection process.

It can be seen that there are f sequential operations involved in the first iterative cycle of any of the two iterative process of system 5 ,

where a sequential operation is here defined to include all the operations involved in the updating process for one component of X (Table 3.3). Each subsequent cycle then requires n sequential operations. The total number of sequential operations involved in the detection process of system 5 is therefore

$$n_s = 2(n)(n_c - 1) + 2f \qquad (3.61)$$

where n_c is the number of iterative cycles used in each of the two iterative processes of system 5, and f is the integer defined in eqn. 3.58 or 3.59.

It can be seen from eqns. 3.41, 3.58, and 3.59 that the initial vector $(X_o)_1$ of **X** used in the first iterative process of system 5 is exactly the same as the initial vector of **X** used in the iterative process of system 3.5. Furthermore, system 5 also uses the iterative process of system 3.5 to operate over the n components of **X** in each of its two iterative processes. It follows that the vector of **X** obtained at the end of the first iterative process of system 5 is always the same as the vector of **X** obtained at the end of the iterative process of system 3.5, for a given vector **R'** and a given number of iterative cycles n_c used in the iterative process. The condition (that is, the inequality of (3.42)) required to achieve **X** = **S** at the end of the iterative process of system 3.5 in the absence of noise, must therefore also hold for the first iterative process of system 5. This means that if the inequality of (3.42) is satisfied, then **X** = **S** at the end of the first iterative process of system 5 so that the corresponding value of $\|\mathbf{R}' - \mathbf{XY}\|$ is zero here, bearing in mind that **R'** = **SY** here. Consequently, this vector of **X** (where **X** = **S**) is always taken as the detected vector **S'** of **S** in system 5 and error-free detection is hence achieved here. Thus, the condition

for system 5 to achieve error-free detection in the absence of noise is the same as that for system 3.5 and is as given by the inequality of (3.42).

Computer simulation tests have been carried out to determine the tolerance to Gaussian noise of system 5 operating over channel A whose sampled impulse response is as given in Table 2.1. The more promising results of these tests are shown in Fig. 3.11. The bit error rate (P_e) and the signal to noise ratio (SNR) here are as defined by eqns. 3.39 and 3.40 respectively. The 95% confidence limits of the results here are about ±0.5 dB. The optimum system here is, of course, the system described in section 3.3.

It can be seen from Figs. 3.10 and 3.11 that, system 5 performs much better than system 4. Furthermore, the tolerance to noise of system 5 also appears to approach that of the optimum system at high signal to noise ratios where the error rates drop below 10^{-4}. The number of sequential operations (eqn. 3.61) required in the detection process of system 5 with $n = 2$ and $f = 3$ is only 22 which is a small number. System 5 therefore appears to be very promising when operating over channel A here.

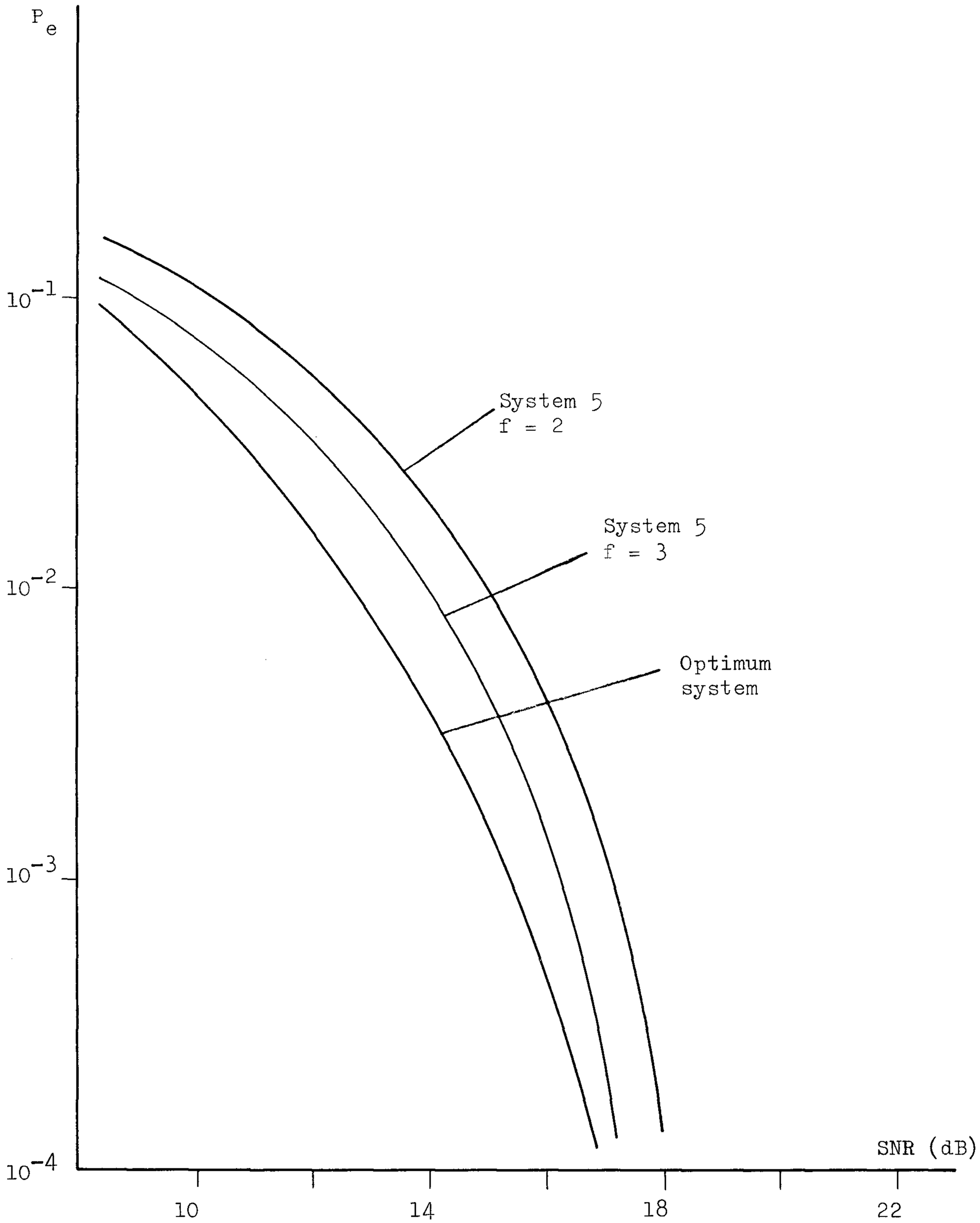


Fig. 3.11 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 5 operating over channel A (Table 2.1).
Number of iterative cycles $n_c = 2$.
f : integer defined in eqns. 3.58 and 3.59.

3.10 System 6

System 6 is a further development of system 5 in that, it uses three separate but similar iterative processes (instead of two) in the detection of the data symbol s_1. Each of these three iterative processes employs the same piece of equipment shown in Fig. 3.4. The first two iterative processes here are exactly the same as the two iterative processes used in system 5. The third iterative process uses the same initial vector of the n-component vector **X** used in any of the systems 3.1 - 3.4.

Thus, in each of the first two iterative processes of system 6, the initial vector X_o, of **X**, is first set to the vector given by eqn. 3.58 (for the first iterative process) or eqn. 3.59 (for the second iterative process). The initial vector of R_a (eqn. 3.22) is then evaluated as $R' - X_oY$. The iterative process of system 3.5 is then applied here, starting from the updating process for x_1 to that for x_n for each iterative cycle (except the first iterative cycle where the iterative process operates from x_{n-f+1} to x_n), using the arrangement shown in Fig. 3.4. At the end of the iterative process when the number of iterative cycles exceeds the preset threshold value n_c, the quantity $\|R' - XY\|$ is measured and stored together with the corresponding vector of **X**. The whole process just described is then repeated for the third iterative process but now the initial vector of **X** is set to that given by eqn. 3.37, and the first iterative cycle now involves only the updating process for x_n. This third iterative process is otherwise the same as any of the first two iterative processes, and at the end of the iterative process, the quantity $\|R' - XY\|$ is measured and stored together with corresponding vector of **X**. The detected vector S' of S is now taken to be the vector of **X** associated with the smallest value of $\|R' - XY\|$. However, only the first component of S' is accepted as the detected value of the data symbol s_1

The last n-1 components $s'_2, s'_3, \cdots, s'_n$ of S' are then shifted one place to the front and used as the initial values of the corresponding n-1 components $x_1, x_2, \cdots, x_{n-1}$ of X for the next detection process.

It can be seen from the detection process just described that, the total number of sequential operations involved here is

$$n_s = 2(n)(n_c - 1) + f + (n)(n_c - 1) + 1 \qquad (3.62)$$

where n_c is the number of iterative cycles used in each of the three iterative processes here, and f is the integer defined in eqn. 3.58 or 3.59. A sequential operation here is defined to include all the operations involved in the updating process for one component of X (Table 3.3).

Following the same arguments given in appendix A2, it can be shown that in the absence of noise, the vector of X obtained at the end of the third iterative process here is always equal to the data-symbol vector S, bearing in mind that this iterative process uses the same initial vector of X as that used in the iterative process of any of the systems 3.1 - 3.4. Consequently, the quantity $\|R' - XY\|$ associated with the third iterative process of system 6 is always equal to zero here, and the corresponding vector of X (where X = S) is hence always taken as the detected vector S' of S here. This means that, system 6 is always able to achieve error-free detection in the absence of noise.

Computer simulation tests have been carried out to determine the tolerance to Gaussian noise of system 6 operating over channel A (Table 2.1), and the results are shown in Fig. 3.12. The 95% confidence limits of these results are about $\pm$0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and 3.40 respectively. The optimum system shown in Fig. 3.12 is the system described in section 3.3.

It can be seen from Figs. 3.11 and 3.12 that, system 6 has a very similar tolerance to noise as that of system 5 when operating over channel A (Table 2.1). The inclusion of the third iterative process in system 6 therefore does not seem to improve its performance here. Nevertheless, system 6 requires more sequential operations than that required in system 5, as can be seen from eqns. 3.61 and 3.62. System 6 therefore appears to be less preferable than system 5 when operating over channel A , since it requires more sequential operations but has about the same tolerance to noise as that of system 5.

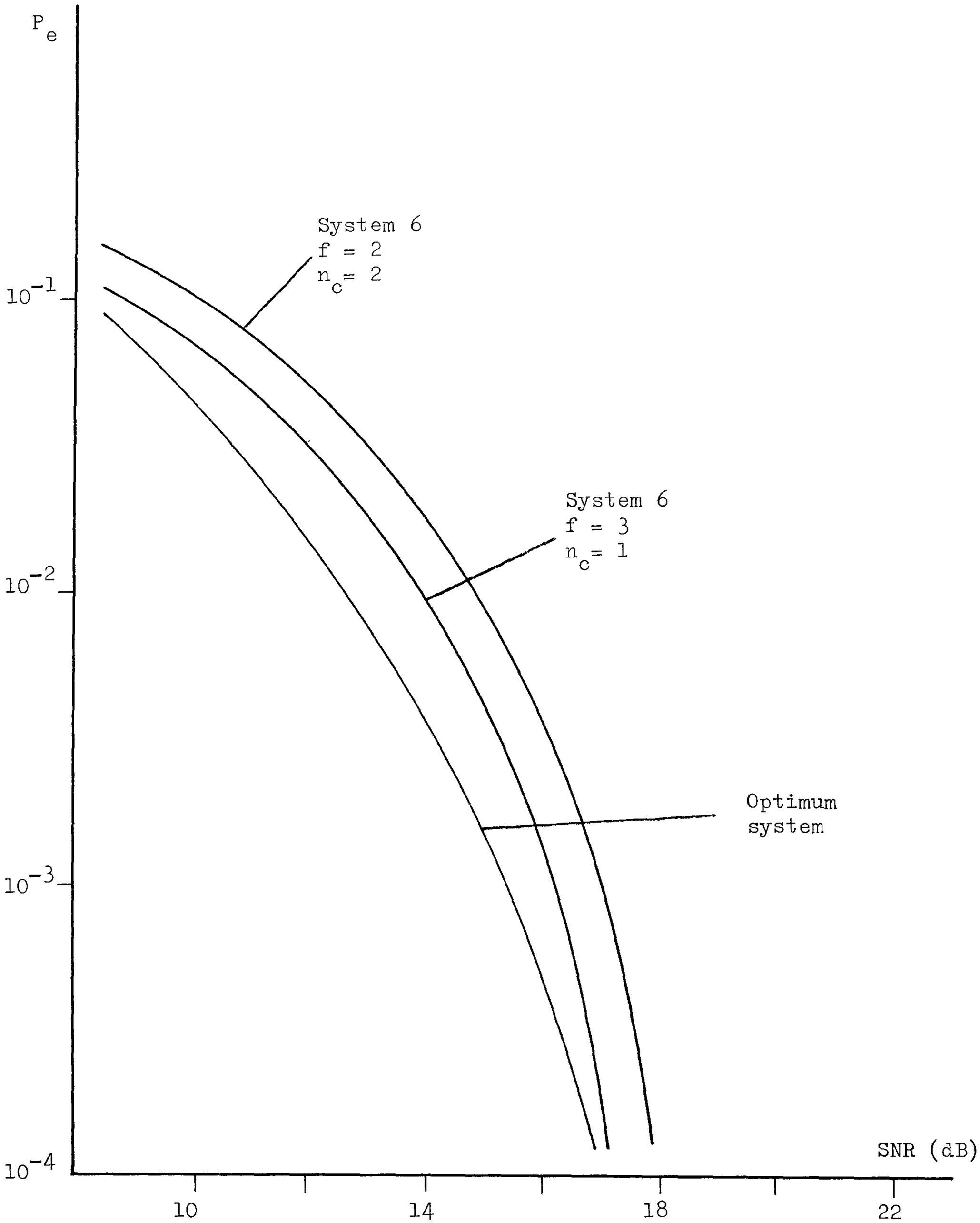


Fig. 3.12 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 6 operating over channel A (Table 2.1).

n_c : number of iterative cycles.

f : integer defined in eqns. 3.50 and 3.51.

3.11 System 7

System 7 is a further development of system 6 in that, it uses four separate but similar iterative processes (instead of three) in the detection of the data symbol s_1. Each iterative process here again uses the same piece of equipment shown in Fig. 3.4. The first three iterative processes here are the same as the respective three iterative processes of system 6. The fourth iterative process, however, uses the following initial vector for X (eqn. 3.13)

$$X_o = \begin{bmatrix} -s'_2 & -s'_3 & \cdots & -s'_n & 0 \end{bmatrix} \qquad (3.63)$$

where $s'_2, s'_3, \cdots, s'_n$ are the detected values of the corresponding data symbols obtained at the end of the previous detection process. This initial vector of X will be further discussed shortly.

Thus, in each of the first two iterative processes of system 7, the initial vector of X is first set to the vector given by eqn. 3.58 (for the first iterative process) or eqn. 3.59 (for the second iterative process), and the initial vector of R_a (eqn. 3.22) is then set to the corresponding vector of R' - XY here. The iterative process of system 3.5 is then applied here, starting from the updating process for x_1 to that for x_n for each iterative cycle (except the first iterative cycle where the iterative prccess operates from x_{n-f+1} to x_n), using the arrangement shcwn in Fig. 3.4. At the end of the iterative prccess (when the number of iterative cycles exceeds the preset threshold value n_c), the quantity $\|R' - XY\|$ is measured and stored together with the corresponding vector of X. Each of the third and fourth iterative processes is the same as any of the first two iterative processes just described except that, the initial vector of X is here set to that given by eqn. 3.37 (for the third iterative process) or eqn. 3.63 (for the fourth iterative

process), and the first iterative cycle here involves only the updating process for x_n. There are now four vectors of $\{X\}$ obtained at the end of the four iterative processes here, and the detected vector S' of S is now taken as the vector of X associated with the smallest value of $\|R' - XY\|$. However, only the first component of S' is accepted as the detected value of the data symbol s_1. The last n-1 components $s'_2, s'_3, \cdots, s'_n$ of S' are then shifted one place to the front and used as the initial values of the corresponding n-1 components $x_1, x_2, \cdots, x_{n-1}$ of X for the next detection process.

It can be seen from the descriptions given above that, the total number of sequential operations involved in the detection process of system 7 is

$$n_s = 2(n)(n_c - 1) + f + 2(n)(n_c - 1) + 1 \qquad (3.64)$$

where n_c is the number of iterative cycles used in each of the three iterative processes here, and f is the integer defined in eqn. 3.58 or 3.59. A sequential operation here is defined to include all the operations involved in the updating process for one component of X.

It has been shown before that, in the absence of noise, the vector of X obtained at the end of the third iterative process here is always equal to the data-symbol vector S and is always taken as the detected vector of S, bearing in mind that this iterative process is exactly the same as the third iterative process used in system 6. It therefore follows that, system 7 is always able to achieve error-free detection in the absence of noise. The fourth iterative process of system 7 is selected based on the following reason. In the detection of the data-symbol vector S, the iterative process here operates on the n-component vector R' to

determine, in turn, the values of the n components $s_1, s_2, \cdots, s_n$ of S. When a component of S is incorrectly determined, its intersymbol interference in the determination of another component of S is doubled instead of being removed. Thus, if the detected value s_2', of s_2, is determined with error, then it is very likely that this error will propagate so that some of the components $s_3', s_4', \cdots, s_n'$ of the detected vector S' of S are now different from the corresponding components of S. In the special case when all the values of $s_2, s_3, \cdots, s_n$ are detected with errors, so that $s_2' = -s_2, s_3' = -s_3, \cdots, s_n' = -s_n$, then the first n-1 components of the initial vector of X used in the fourth iterative process of system 7 (eqn. 3.63) become exactly the same as the corresponding components of S. In this case, it is very likely for this iterative process to obtain the vector X as the possible vector of S associated with the minimum distance $\|R' - SY\|$, since the iterative process here is non-divergent. It is, of course, not very likely that all the components of S are detected with errors at high signal to noise ratios. However, it appears intuitively that so long as most of the components of S are detected with errors at the end of a detection process, then the fourth iterative process of system 7 is likely to obtain, in the next detection process, a vector of X that has the smallest value of $\|R' - XY\|$ as compared to those obtained by the first three iterative processes of system 7. Consequently, the tolerance to noise of the system may be improved by the inclusion of this fourth iterative process which uses the n-component vector given by eqn. 3.63 as the initial vector of X.

Computer simulation tests have been carried out to determine the tolerance to Gaussian noise of system 7 operating over channel A

(Table 2.1), and the results of these tests are shown in Fig. 3.13. The bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and .340 respectively. The 95% confidence limits of these results are about ± 0.5 dB. The performance of the optimum system described in section 3.3 is also included here.

As Figs. 3.11 and 3.13 show, there is no significant difference between the tolerance to noise of system 7 and that of system 5. The inclusion of the additional third and fourth iterative processes in system 7 therefore appears to be ineffective over channel A (Table 2.1).

Clearly, the use of more than four iterative processes in the detection process for s_1 will further increase the number of sequential operations involved here. The maximum possible improvement in the tolerance to noise to be gained over system5 is, however, very small for channel A (Table 2.1) as is shown in Fig. 3.11. As a result of this, no further investigations have been carried out to study the effects of using more iterative processes in the detection process for s_1.

The following systems in this chapter differ from systems 1 - 7 in that the networks $\{Y_h^T\}$ shown in Fig. 3.4 are now replaced by some other networks $\{Z_h^T\}$ for the iterative process used in each of these systems. Two different sets of $\{Z_h^T\}$ have been developed and studied, and are described in the following sections. The modified block diagram of Fig. 3.4 is as shown in Fig. 3.14. The basic operations of these systems are otherwise similar to those of the systems 1 - 7.

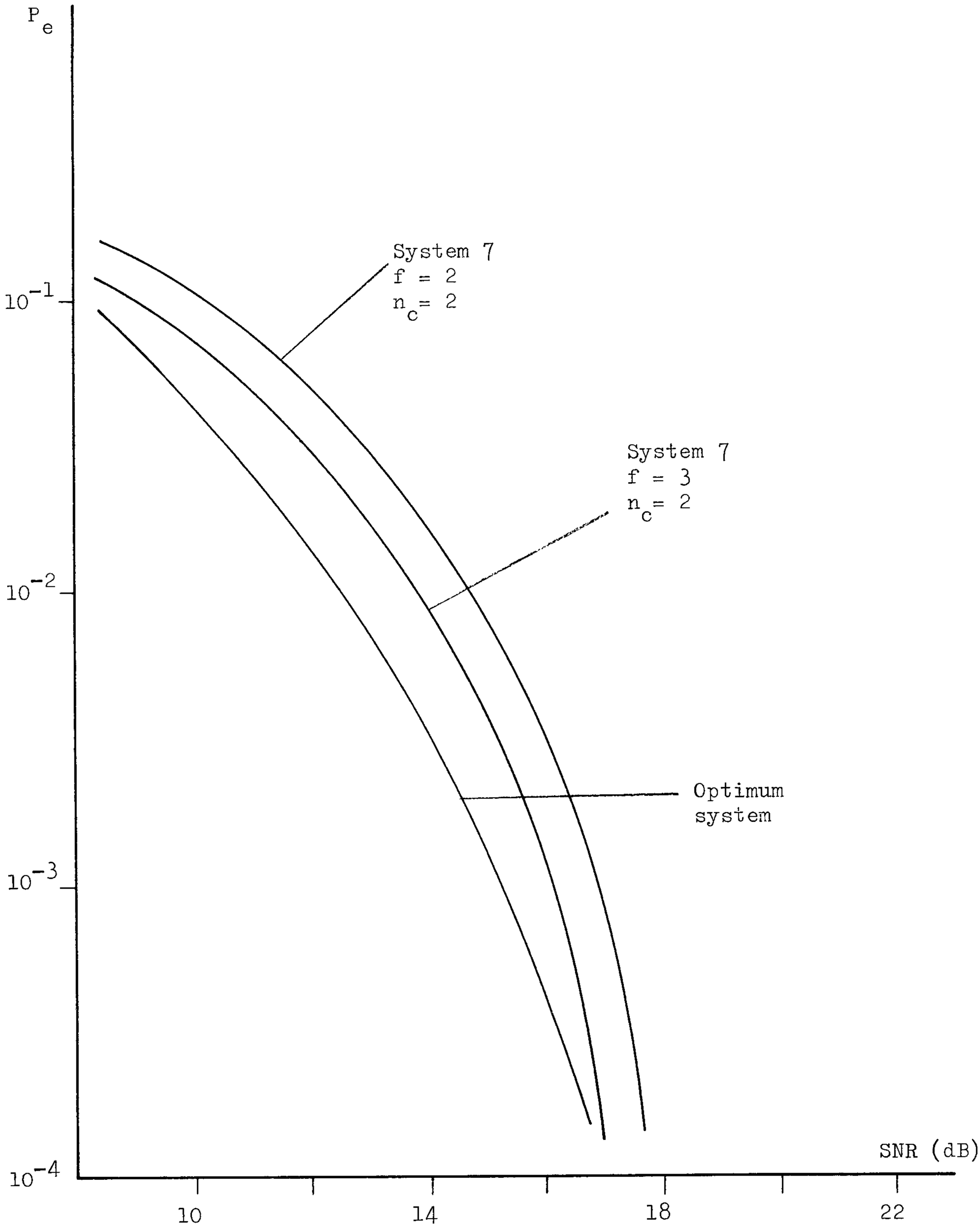


Fig. 3.13 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 7 operating over channel A (Table 2.1).

n_c : number of iterative cycles.

f : integer defined in eqns.3.58 and 3.59.

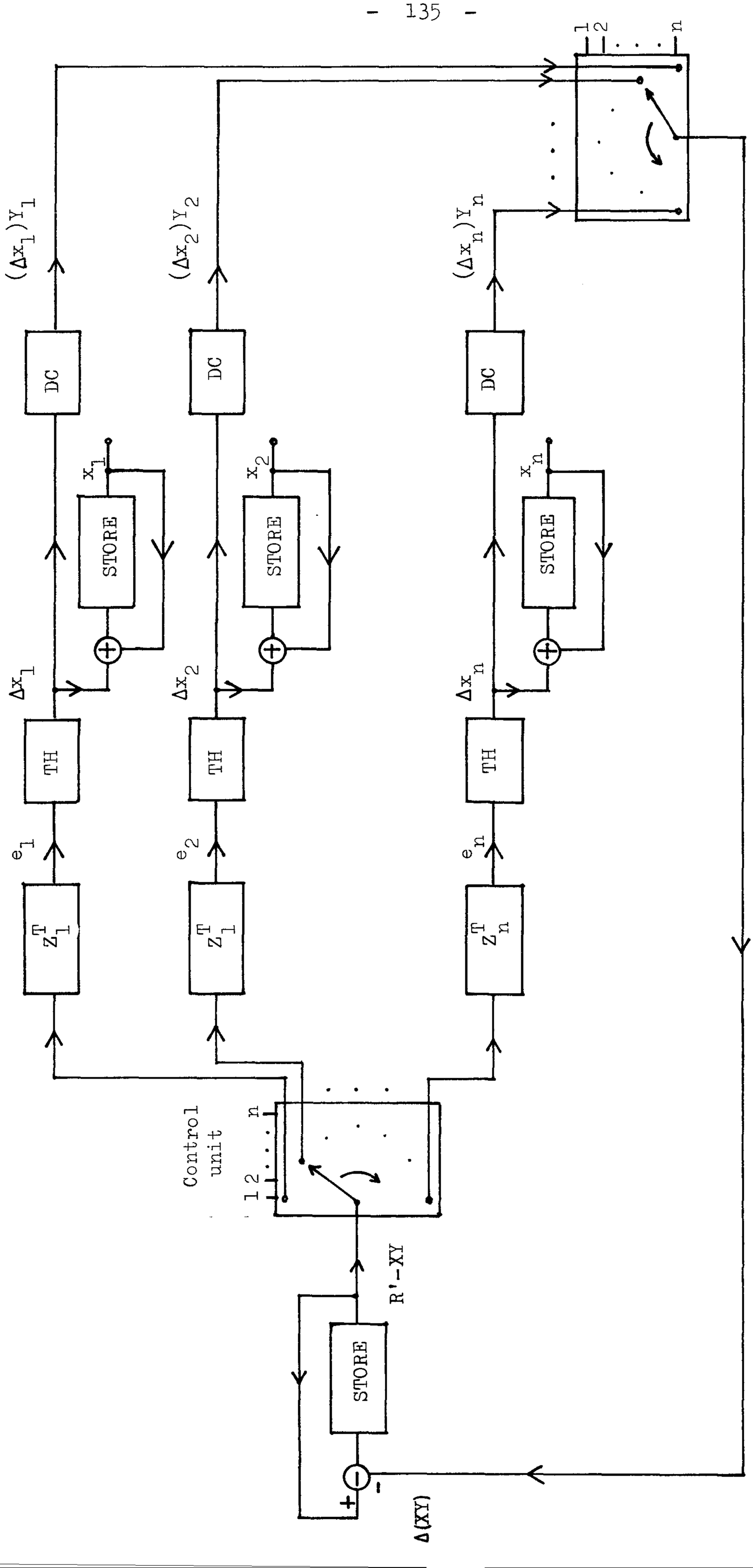


Fig. 3.14 Iterative process used in any of the systems 8 - 11.

TH : threshold detector

DC : decision device

3.12 System 8

System 8 is a modification of system 1. The modification here is such that the iterative process to be used in the system now employs the arrangement shown in Fig. 3.14 which is obtained from Fig. 3.4 by replacing the networks $\{Y_h^T\}$ by some other networks $\{Z_h^T\}$. The n n-component vectors $Z_1, Z_2, \cdots, Z_n$ corresponding to the n networks $\{Z_h^T\}$ of Fig. 3.14 are here derived from the n n-component vectors $Y_1, Y_2, \cdots, Y_n$ by using the Gram-Schmidt orthogonalisation process,(E5) where the $\{Y_h\}$ are as defined by eqn. 3.9. It is reminded that Z_h^T is the transpose of the row matrix Z_h and that a row matrix is also treated as a vector in this thesis. Thus, a set of n n-component vectors is first evaluated in the orthogonalisation process as

$$
\begin{aligned}
Z_1' &= Y_1 \\
Z_2' &= Y_2 - \frac{1}{\|Z_1'\|^2} Y_2 (Z_1')^T Z_1' \\
&\vdots \\
Z_n' &= Y_n - \frac{1}{\|Z_{n-1}'\|^2} Y_n (Z_{n-1}')^T Z_{n-1}' - \cdots - \frac{1}{\|Z_1'\|^2} Y_n (Z_1')^T Z_1'
\end{aligned}
\qquad (3.65)
$$

where $\|Z_h'\|$ is the Euclidean norm or length of the vector Z_h'. The n n-component vectors $Z_1, Z_2, \cdots, Z_n$ are then evaluated as

$$
Z_h = \frac{Z_h'}{\|Z_h'\|} \qquad (3.66)
$$

for $h = 1, 2, \cdots, n$. Clearly, $Z_1, Z_2, \cdots, Z_n$ are all unit vectors

so that

$$Z_h(Z_h)^T = \| Z_h \|^2 = 1 \qquad (3.67)$$

for $h = 1, 2, \cdots, n$. Furthermore, the orthogonalisation process of eqn. 3.65 necessarily ensures that[E5]

$$Z'_h(Z'_j)^T = 0 \qquad \text{for} \quad h \neq j \qquad (3.68)$$

so that from eqn. 3.66,

$$Z_h(Z_j)^T = 0 \qquad \text{for} \quad h \neq j \qquad (3.69)$$

This means that, the n vectors $Z_1, Z_2, \cdots, Z_n$ derived here form a set of orthonormal vectors. Another interesting property of these $\{Z_h^T\}$ can be derived from eqn. 3.65 (appendix A4) and is

$$\begin{aligned} Y_j(Z_h)^T &= 0 \qquad \text{if} \quad j < h \\ &> 0 \qquad \text{if} \quad j = h \end{aligned} \qquad (3.70)$$

for $h = 1, 2, \cdots, n$. This implies that Z_h is orthogonal to $Y_1, Y_2, \cdots, Y_{h-1}$ and that

$$Y_h(Z_h)^T = \left| Y_h(Z_h)^T \right| > 0 \qquad (3.71)$$

for $h = 1, 2, \cdots, n$. The operation of the detection process of system 8 is now described as below.

In the detection of the data symbol s_1 from R' (eqns. 3.5 - 3.9), system 8 uses an iterative process that has the arrangement shown in Fig. 3.14. The n-component vector R_a at the input to the networks $\{Z_h^T\}$ is as defined by eqn. 3.22 to be R' - XY, and the n quantities at the n outputs of the corresponding networks $\{Z_h^T\}$ are here given by

$$e_h = (R' - XY)Z_h^T = R_a Z_h^T \qquad (3.72)$$

for $h = 1, 2, \cdots, n$ where X is the n-component vector defined by eqn. 3.13, and Y is the $n \times n$ matrix defined by eqn. 3.9. The $\{Z_h\}$ here are of course the n n-component vectors derived by using eqns. 3.65 and 3.66, bearing in mind that Z_h^T is the transpose of Z_h. Two different arrangements of this iterative process are investigated in systems 8.1 and 8.2, and are described as below.

System 8.1

The n components $x_1, x_2, \cdots, x_n$ of the vector X shown in Fig. 3.14 are initially set to zero, so that the initial vector of R_a (eqn. 3.22) now becomes the n-component vector R' (eqn. 3.7). Having set the n components of X and R_a to their respective initial values, the iterative process of system 8.1 now operates as follow. The value of e_1 (eqn. 3.72) is first evaluated at the output of the network Z_1^T shown in Fig. 3.14. This value of e_1 is then compared with its associated thresholds $\pm t_1$ to give the value of Δx_1, where t_1 is a positive quantity to be described shortly, and Δx_1 is the increment involved in the value of x_1 (eqn. 3.26). The value of Δx_1 is here determined according to the decision rule given in Table 3.1. The value of x_1 is then updated (adjusted) by using eqn. 3.30, as the sum of its previous value and the value of Δx_1. The vector R_a is then updated by removing from it all components of $(\Delta x_1 Y_1)$ (eqn. 3.31). Having updated x_1 and R_a, the value of e_2 is next evaluated at the output of the network Z_2^T shown in Fig. 3.14. The value of Δx_2 is then determined according to Table 3.1. The value of x_2 and the vector R_a are then updated by using eqns. 3.30 and 3.31 respectively. The iterative process continues to operate in this way with the sequential adjustment (updating) of x_3 to x_n,

using the arrangement of Fig. 3.14. This completes the first iterative cycle of the iterative process here. Each subsequent iterative cycle operates in the same way as for the first iterative cycle except that Table 3.2 (instead of Table 3.1) is now used to determine the value of Δx_h $(1 \leq h \leq n)$ in the updating process for x_h.

In the iterative process of system 8.1 just described, a counter is used to count the number of iterative cycles. When the counter exceeds a given threshold n_c, the iterative process is terminated and the value of x_1 is then taken as the detected value of the data symbol s_1.

System 8.2

System 8.2 is only slightly different from system 8.1 in that, the iterative process here now begins with the updating process for x_n and works towards x_1 for each iterative cycle. This system is otherwise the same as system 8.1 which operates from x_1 to x_n for each iterative cycle.

Four different versions of the values of $\{t_h\}$ to be used to determine the corresponding values of $\{\Delta x_h\}$ (Table 3.1 or 3.2) in the iterative process of each of the systems 8.1 and 8.2 are considered and are as given in Table 3.8.

It is shown in appendix A5 that, divergence can occur in the iterative process of any of the systems 8.1 and 8.2, so long as the networks $\{Z_h^T\}$ shown in Fig. 3.14 are not the same as the networks $\{Y_h^T\}$ used in Fig. 3.4. Divergence is defined to have occurred when the Euclidean distance $\|R' - XY\|$ increases during the iterative process, bearing in mind that, under the various conditions assumed here, the vector of X (being here a possible vector of S) that has the minimum

value of $\|R' - XY\|$ is the possible vector of S most likely to be correct in the detection process for S. Nevertheless, it can be shown (appendix A6) that, so long as the number of iterative cycles n_c used in the iterative process of any of the systems 8.1 and 8.2 is sufficiently large, then error-free detection in the absence of noise can always be achieved by any of the systems 8.1 and 8.2.

Versions	1st Cycle	2nd Cycle	All Subsequent Cycles
a	0	$\lvert Y_h(Z_h^T)\rvert$	$\lvert Y_h(Z_h^T)\rvert$
b	$\frac{1}{4}\lvert Y_h(Z_h^T)\rvert$	$\lvert Y_h(Z_h^T)\rvert$ if $x_h=\pm 1$ 0 if $x_h= 0$	$\lvert Y_h(Z_h^T)\rvert$
c	$\frac{1}{2}\lvert Y_h(Z_h^T)\rvert$	$\lvert Y_h(Z_h^T)\rvert$ if $x_h=\pm 1$ 0 if $x_h= 0$	$\lvert Y_h(Z_h^T)\rvert$
d	$\lvert Y_h(Z_h^T)\rvert$	$\lvert Y_h(Z_h^T)\rvert$ if $x_h=\pm 1$ 0 if $x_h= 0$	$\lvert Y_h(Z_h^T)\rvert$

Table 3.8 Four different versions of threshold values $\{t_h\}$, for $h = 1, 2, \cdots, n$, to be used to determine the values of $\{\Delta x_h\}$ (Table 3.1 or 3.2) in the iterative process of each of the systems 8.1 and 8.2.

It can in fact be shown that the tolerance to noise of system 8.1 is the same as that of system 8.2 when a sufficiently large number of iterative cycles is used in each of these systems. This is given as follow. It can be seen from eqns. 3.8 and 3.72 that, the value of e_h used in the updating process for x_h here may be reduced to

$$\begin{aligned} e_h &= (SY - XY)Z_h^T + WZ_h^T \\ &= (s_1 - x_1)Y_1Z_h^T + (s_2 - x_2)Y_2Z_h^T + \cdots \\ &\qquad + (s_n - x_n)Y_nZ_h^T + w_h' \end{aligned} \tag{3.73}$$

for $h = 1, 2, \cdots, n$ where w_h', being equal to WZ_h^T, is the noise component here. Since the components of W (eqn. 3.6) are Gaussian random variables with zero mean and variance σ^2, it can be shown[A9] that w_h' is also a Gaussian random variable with zero mean and variance $\sigma^2\|Z_h\|^2$ or σ^2, bearing in mind that $\|Z_h\| = 1$ here (eqn. 3.67). By using the property of (3.70), eqn. 3.73 can be further reduced to

$$\begin{aligned} e_h &= (s_h - x_h)\left|Y_hZ_h^T\right| + (s_{h+1} - x_{h+1})Y_{h+1}Z_h^T + \cdots \\ &\qquad + (s_n - x_n)Y_nZ_h^T + w_h' \end{aligned} \tag{3.74}$$

for $h = 1, 2, \cdots, n$ where $\left|Y_hZ_h^T\right|$ is the absolute value of the scalar quantity $Y_hZ_h^T$. Clearly, the value of e_h here is independent of the values of $x_1, x_2, \cdots, x_{h-1}$. It follows that the set of orthonormal vectors $\{Z_h\}$ (eqns. 3.65 and 3.66) used here effectively removes from e_h the intersymbol interference components associated with $s_1, s_2, \cdots, s_{h-1}$, bearing in mind that the detected value of s_h is here determined as the the value of x_h from e_h. In the following analysis, it is more convenient if eqn. 3.74 is rearranged to become

$$\begin{aligned} e_h &= (-x_h)\left|Y_hZ_h^T\right| + s_h\left|Y_hZ_h^T\right| + \sum_{k=h+1}^{n}(s_k - x_k)Y_kZ_h^T + w_h' \\ &= (-x_h)\left|Y_hZ_h^T\right| + v_h \end{aligned} \tag{3.75}$$

where

$$v_h = s_h|Y_h Z_h^T| + \sum_{k=h+1}^{n}(s_k - x_k)Y_k Z_h^T + w_h' \qquad (3.76)$$

for $h = 1, 2, \cdots, n$. Two cases may now occur in updating the value of x_h from e_h here. In the first case, the values of $s_h, s_{h+1}, \cdots, s_n, x_{h+1}, x_{h+2}, \cdots, x_n$, and w_h' are such that

$$v_h > 0 \qquad (3.77)$$

and in the second case, these values are such that

$$v_h \leq 0 \qquad (3.78)$$

where v_h is, of course, as defined in eqns. 3.75 and 3.76. Assuming that the value of x_h is here updated by using Table 3.2 which is used in the second and subsequent iterative cycles of the iterative process of any of the systems 8.1 and 8.2. Thus, it can be seen from eqns. 3.30, 3.75, and Tables 3.2, 3.8 that, the values of x_h and e_h before and after the updating of x_h for the two cases considered here (eqns. 3.77 and 3.78) can be summarised in Table 3.9. It can now be seen from Table 3.9 that, the updated value of x_h is entirely determined by the value of d_h except for the case when $x_h = 1$ and $v_h = 0$. However, the case of $v_h = 0$ is very unlikely to occur and is thus neglected in the analysis here. Thus, if $v_h > 0$, then the updated value of x_h is always equal to 1, and if $v_h \leq 0$, then the updated value of x_h is always equal to -1. Furthermore, it can be seen from eqn. 3.75 that, given the values of $s_h, s_{h+1}, \cdots, s_n$ and w_h', the value of v_h is entirely determined by the values of $x_{h+1}, x_{h+2}, \cdots, x_n$. It therefore follows that, given the values of $s_h, s_{h+1}, \cdots, s_n$ and w_h', the updated

value of x_h (Table 3.9) is also entirely determined by the values of x_{h+1}, x_{h+2}, $\cdot\cdot$, x_n. Clearly, given the values of s_n and w'_n , the updated value of x_n here must always remain to be unchanged in the subsequent updating processes for x_n because the value of v_n, being equal to $s_n|Y_n Z_n^T| + w'_n$, is always unchanged during the iterative process here. For convenience, the value of x_h that remains to be unchagned in the subsequent updating processes for x_h is here referred to as the steady-state value of x_h, for $h = 1, 2, \cdot\cdot , n$. Thus, having set the value of x_n to its steady-state value, the next updated value of x_{n-1} (Table 3.9) must now always be equal to the steady-state value of x_{n-1}, because the updated value of x_{n-1} is here determined entirely by the value of x_n. Similarly, having set the values of x_n and x_{n-1} to their respective steady-state values, the next updated value of x_{n-2} must now always be equal to the steady-state value of x_{n-2}, since the updated value of x_{n-2} is here determined entirely by the values of x_n and x_{n-1}. This argument can, of course, be applied to the updated values of x_{n-3}, x_{n-4}, $\cdot\cdot$, x_1. Consequently, all the values of x_1, x_2, $\cdot\cdot$, x_n will eventually be set to their respective steady-state values regardless of what their initial values (before the iterative process begins) are and whether the iterative process here operates from x_1 to x_n or from x_n to x_1 for each iterative cycle. This obviously means that the various versions of systems 8.1 and 8.2 must always have the same tolerance to noise in the detection process for s_1, so long as a sufficiently large number of iterative cycles is used here.

Thus, after the number of iterative cycles used in the iterative process here has exceeded a certain threshold value, all the values of x_1, x_2, $\cdot\cdot$, x_n will have been set to their respective steady-state values and will from then on remain unchanged in the subsequent iterative

cycles. The maximum number of iterative cycles required to set all the values of $x_1, x_2, \cdots, x_n$ to their respective steady-state values can be derived from the arguments described above for each of the various versions of systems 8.1 and 8.2, and is as given in Table 3.10. For example, when system 8.1 with version d (Table 3.8) is considered, Tables 3.2, 3.8, and 3.9 suggest that the value of x_n is set to its steady-state value by the end of the second iterative cycle. The updated value of x_{n-1} in the third iterative cycle is therefore always equal to its steady-state value since the value of x_n is now the steady-state value of x_n, bearing in mind that the updated value of x_h $(1 \leq h \leq n)$ is here determined entirely by the values of $x_{h+1}, x_{h+2}, \cdots, x_n$. Similarly, the updated value of x_{n-2} in the fourth iterative cycle must always be equal to its steady-state value since the values of x_n and x_{n-1} are now equal to their respective steady-state values. The argument just described can be applied to $x_{n-3}, x_{n-4}, \cdots, x_1$ and consequently, all the values of $x_1, x_2, \cdots, x_n$ must have been set to their respective steady-state values at the end of the (n+1)th iterative cycle as is shown in Table 3.10. Thus, the tolerance to noise of system 8.1 with version d will not be changed by increasing the number of iterative cycles used in the iterative process of the given system beyond the value of n+1. It is, of course, possible for all the values of $x_1, x_2, \cdots, x_n$ to be set to their respective steady-state values before the number of iterative cycles reaches that given in Table 3.10. Fig. 3.15 shows the number of errors in the detection of 100 transmitted data symbols $\{s_i\}$ in the absence of noise, for the various versions of system 8.1 operating over channel A (Table 2.1). As it appears, the minimum number of iterative cycles required to be used in the iterative process of any of the various versions of system 8.1

to give error-free detection here is 7 which is smaller than the corresponding value given in Table 3.10, as can be expected from the analysis described above.

Computer simulation tests have been carried out to determine the tolerances to Gaussian noise of the various versions (Table 3.8) of systems 8.1 and 8.2 operating over channel A (Table 2.1), assuming the data-transmission system shown in Fig. 2.1. The results of the simulation tests here are shown in Fig. 3.16. The 95% confidence limits of the results shown in Fig. 3.16 are about ± 0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and 3.40 respectively. The optimum system here is, of course, the system described in section 3.3.

The results shown in Fig. 3.16 obviously agree with the analysis given previously. That is, all the various versions of systems 8.1 and 8.2 have the same tolerance to noise. The tolerance to noise of the system employing the iterative process shown in Fig. 3.14 where the $\{Z_h^T\}$ are derived by using eqns. 3.65 and 3.66 is therefore independent of whether the iterative process operates from x_1 to x_n or from x_n to x_1. However, as Fig. 3.16 shows, the tolerance to noise of any of the systems 8.1 and 8.2 is very much poorer than that of the optimum system. One possible reason to this poor performance of any of the systems 8.1 and 8.2 is given as follows. It is recalled that the iterative process of the system here is such that, for given values of $s_1, s_2, \cdots, s_n$, and $w_1', w_2', \cdots, w_n'$, the steady-state value of x_h is entirely determined by the steady-state values of $x_{h+1}, x_{h+2}, \cdots, x_n$, for $h = 1, 2, \cdots, n$. The value of x_h is here determined from the value of e_h given by eqn. 3.74. Clearly, in determining the value of x_h from e_h here, the probalility of x_h being set to the correct value (that is, $x_h = s_h$) is dependent

not only on the magnitude of the signal component $s_h Y_h Z_h^T$, but also on the probabilities of $x_{h+1}, x_{h+2}, \cdots, x_n$ being equal to the corresponding values of $s_{h+1}, s_{h+2}, \cdots, s_n$, for a given noise component w_h'. It therefore appears that, for a given set of noise components $\{w_h'\}$, the probability of s_1 being equal to its detected value or the steady-state value of x_1 is here likely to be dependent to a large extent on the magnitudes of the corresponding signal components $s_1 Y_1 Z_1^T, s_2 Y_2 Z_2^T, \cdots, s_n Y_n Z_n^T$ in $e_1, e_2, \cdots, e_n$. That is, the larger the magnitudes of these signal components, the lower may be the probability of s_1 being detected with error (where an error here means $x_h \neq s_h$). Furthermore, it seems likely that by eliminating from the e_h of eqn. 3.73 the intersymbol interference components associated with $s_1, s_2, \cdots, s_{h-1}$ to give the e_h of eqn. 3.74 used here, the magnitude of the resultant signal component $s_h Y_h Z_h^T$ of e_h may become very small for a given noise level of w_h', for $h = 1, 2, \cdots, n$. Consequently, the magnitudes of the signal components $s_1 Y_1 Z_1^T, s_2 Y_2 Z_2^T, \cdots, s_n Y_n Z_n^T$ here are so small that the probability of s_1 being detected with error is now very high leading to the poor performance of any of the systems 8.1 and 8.2 shown in Fig. 3.16.

Thus, the various versions of systems 8.1 and 8.2 studied here do not appear to have a useful tolerance to noise when operating over channel A (Table 2.1) despite have the many interesting properties described in this section.

	Previous Values		Threshold (Table 3.2)	Updated Values	
	x_h	e_h	t_h	x_h	e_h
$v_h > 0$	-1	$+\lvert Y_h Z_h^T \rvert$	$\lvert Y_h Z_h^T \rvert$	1	$-\lvert Y_h Z_h^T \rvert$
	0	0	0	1	$-\lvert Y_h Z_h^T \rvert$
	1	$-\lvert Y_h Z_h^T \rvert$	$\lvert Y_h Z_h^T \rvert$	1	$-\lvert Y_h Z_h^T \rvert$
$v_h \leq 0$	-1	$+\lvert Y_h Z_h^T \rvert$	$\lvert Y_h Z_h^T \rvert$	-1	$+\lvert Y_h Z_h^T \rvert$
	0	0	0	-1	$+\lvert Y_h Z_h^T \rvert$
	1	$-\lvert Y_h Z_h^T \rvert$	$\lvert Y_h Z_h^T \rvert$	-1	$+\lvert Y_h Z_h^T \rvert$
	1	$-\lvert Y_h Z_h^T \rvert$	$\lvert Y_h Z_h^T \rvert$	1	$-\lvert Y_h Z_h^T \rvert$

Table 3.9 Values of x_h and e_h before and after the updating of x_h for $h = 1, 2, \cdots, n$ in the second and subsequent iterative cycles of the iterative process of system 8. v_h is as defined by eqns. 3.75 and 3.76.

Versions (Table 3.8)	System 8.1	System 8.2
a	n	1
b	n + 1	2
c	n + 1	2
d	n + 1	2

Table 3.10 Maximum number of iterative cycles required in the iterative process to set all the values of $x_1, x_2, \cdots, x_n$ to their respective steady-state values. $n = 8$ in this thesis.

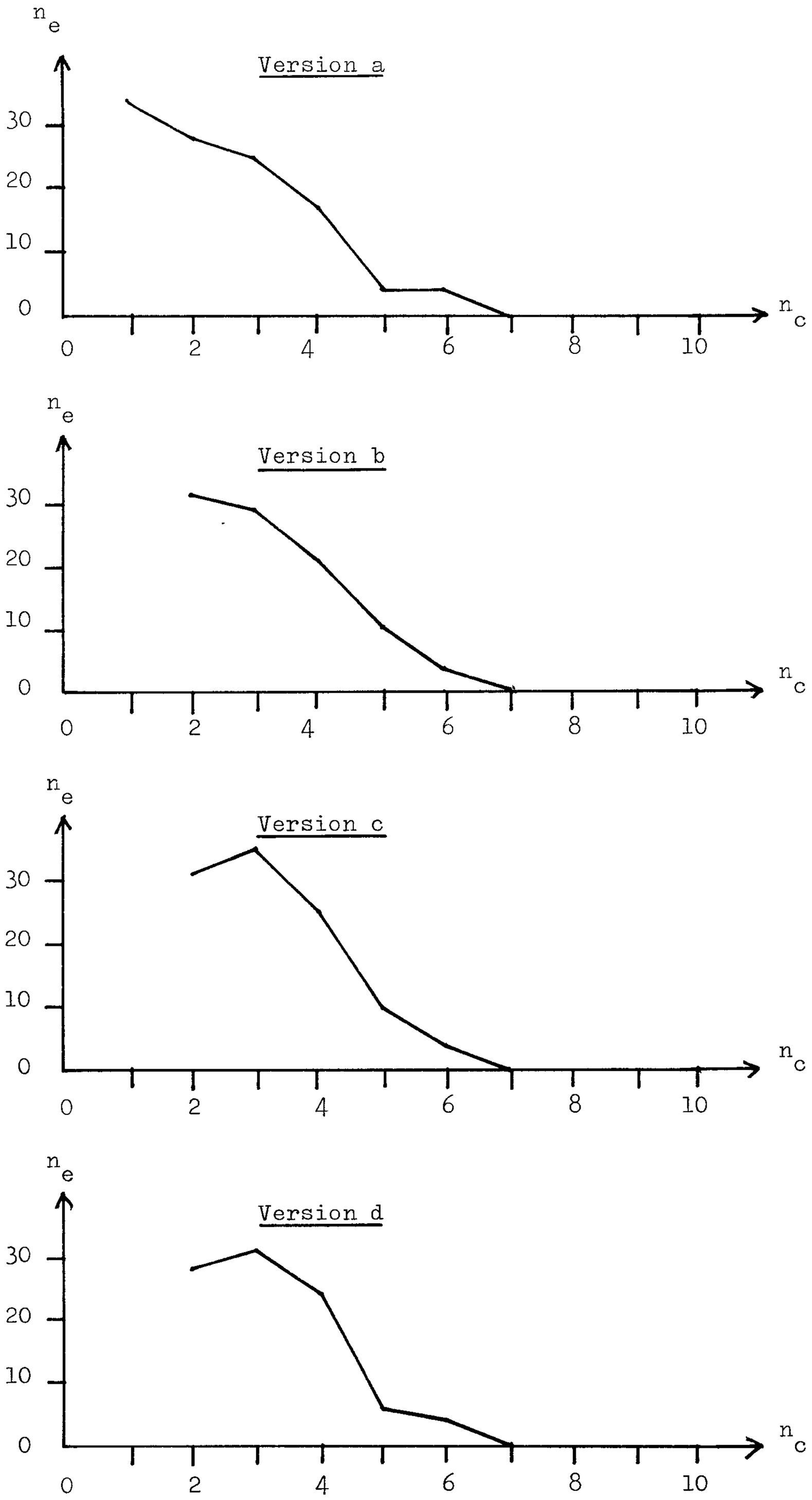


Fig. 3.15 Number of errors (n_e) in the detection of $\{s_i\}$ for system 8.1 with versions a - d (Table 3.8) at various number of iterative cycles (n_c), when 100 data symbols $\{s_i\}$ are transmitted over channel A (Table 2.1) in the absence of noise.

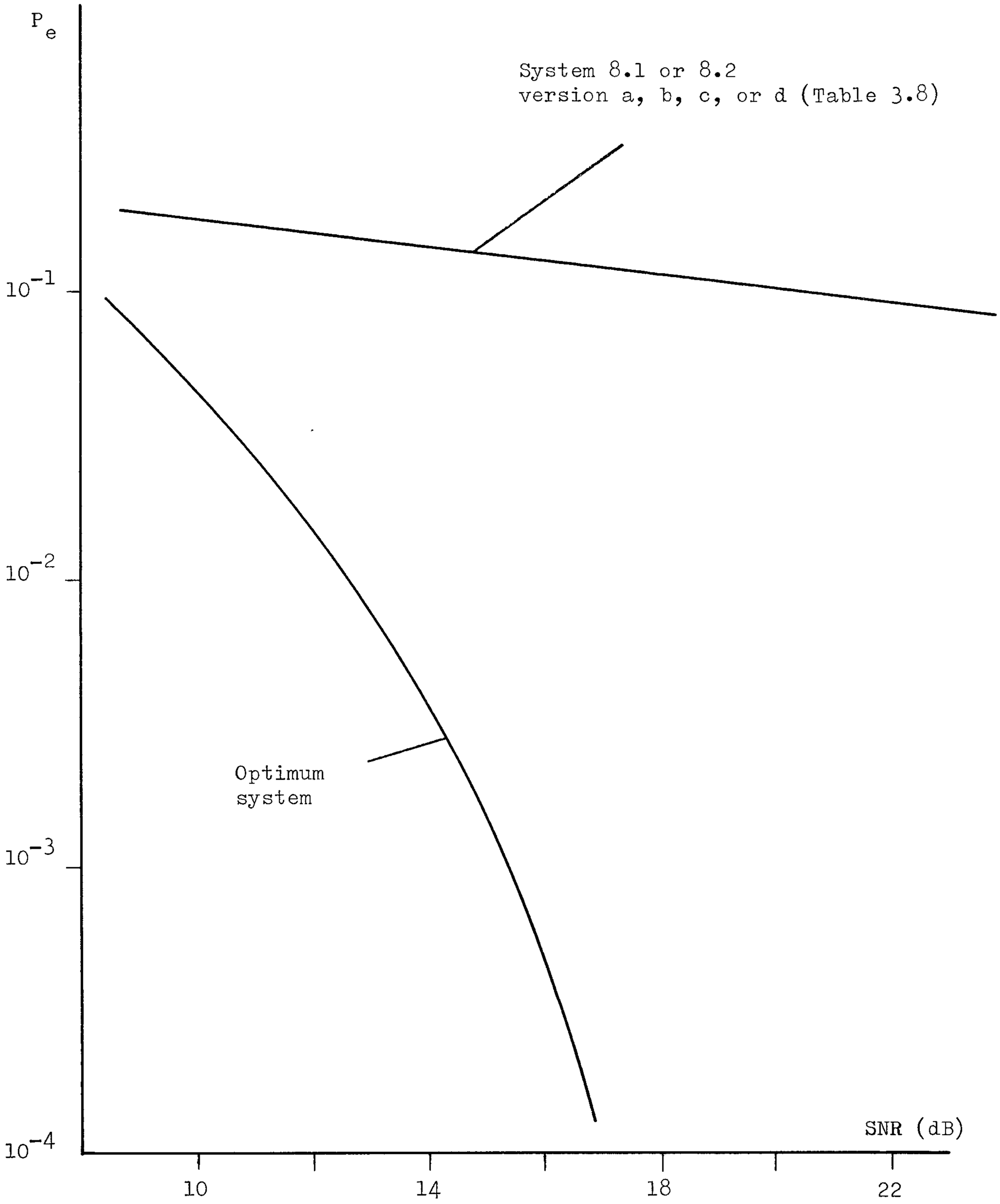


Fig. 3.16 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 8 operating over channel A (Table 2.1).
Number of iterative cycles n_c = 7, 8, 9, or 10 for system 8.1, and n_c = 2, 3, 4, or 5 for system 8.2.

3.13 Systems 9 - 11

The systems to be developed and studied in this section are modifications of system 8. The iterative process used in each of these systems employs the arrangement shown in Fig. 3.14 where the networks $\{Z_h^T\}$ here are not now derived from the Gram-Schmidt orthogonalisation process but are derived as follow.

Before proceeding to derive the networks $\{Z_h^T\}$ to be used in any of the systems 9 - 11, consider first the mechanism that leads to the poor performance (Fig. 3.16) of system 8. In the iterative process of Fig. 3.14, the detected value of the data symbol s_h $(1 \leq h \leq n)$ is determined as the value of x_h from the value of e_h given, in general, by eqn. 3.73. The networks $\{Z_h^T\}$ used in system 8 are such that, the intersymbol interference components associated with $s_1, s_2, \cdots, s_{h-1}$ are removed from e_h so that e_h is now given by eqn. 3.74. However, it appears that by removing these intersymbol interference components from e_h, the magnitude of the signal component $s_h Y_h Z_h^T$ is reduced to be so small (in relation to the magnitude of the noise component w_h') that the overall tolerance to noise of the system now becomes very poor as is shown in Fig. 3.16. It therefore appears that, in order to maximize the tolerance to noise of the system here, the networks $\{Z_h^T\}$ (Fig. 3.14) should be selected to be such that the magnitudes of the intersymbol interference components in e_h are minimized and the magnitude of the signal component in e_h is maximized, for a given noise level in e_h. This is the basis of systems 9 - 11.

Thus, eqn. 3.73 may be rearranged to become

$$e_h = (s_h - x_h) Y_h Z_h^T + I_h + w_h' \qquad (3.79)$$

where

$$I_h = \sum_{\substack{k=1 \\ k \neq h}}^{n} (s_k - x_k) Y_k Z_h^T \qquad (3.80)$$

for $h = 1, 2, \cdots, n$. In eqn. 3.79, $(s_h - x_h) Y_h Z_h^T$ is the signal

component, I_h is the intersymbol interference component, and w'_h is the noise component. In determining the detected value of s_h, as x_h, from the value of e_h here, it is desirable that the relative magnitude of the signal component and the intersymbol interference component in e_h is maximized, for a given noise level. That is, it is desirable to maximize the quantity

$$d_h = |(s_h - x_h)Y_h Z_h^T| - |\sum_{\substack{k=1 \\ k\neq h}}^{n} (s_k - x_k)Y_k Z_h^T| \qquad (3.81)$$

with the constraint that $\|Z_h\| = 1$, for $h = 1, 2, \cdots, n$. The constraint here is to ensure that the noise components $\{w'_h\}$ have a fixed variance σ^2 regardless of what the networks $\{Z_h^T\}$ are, bearing in mind that the $\{w'_h\}$ are Gaussian random variables with zero mean and variance $\sigma^2\|Z_h\|^2$.

Eqn. 3.81 can be rearranged to become

$$d_h = m_h|Y_h Z_h^T| - \sum_{\substack{k=1 \\ k\neq h}}^{n} m_k|Y_k Z_h^T| \qquad (3.82)$$

for $h = 1, 2, \cdots, n$ where the value of m_j is either equal to $(s_j - x_j)$ or $-(s_j - x_j)$. It is, however, very difficult to maximize the quantity given by eqn. 3.82 because it is a nonlinear function of the vector Z_h. Furthermore, since the values of $\{(s_j - x_j)\}$ vary from time to time during the detection process here, the vector Z_h that maximizes the value of d_h here (eqn. 3.82) must also be different during the detection process and this necessarily complicates the operations required in the iterative process here. Consequently, an alternative expression for d_h is proposed in this thesis as d'_h where

$$d'_h = (Y_h Z_h^T)^2 - m_o^2 \sum_{\substack{k=1 \\ k \neq h}}^{n} (Y_k Z_h^T)^2 \qquad (3.83)$$

for $h = 1, 2, \cdots, n$ and the same constant value m_o^2 is now used for all the $\{Z_h\}$ and for the whole of the detection process here. The constant m_o^2 will be further discussed shortly.

Thus, the n-component vector Z_h to be used in the iterative process (Fig. 3.14) of any of the systems 9 - 11 is here selected to maximize the value of d'_h given by eqn. 3.83 for a given value of m_o^2 and with the constraint that $\| Z_h \| = 1$ (or $Z_h Z_h^T = 1$). Eqn. 3.83 can be rearranged to become

$$\begin{aligned} d'_h &= Z_h \left(Y_h^T Y_h - m_o^2 \sum_{\substack{k=1 \\ k \neq h}}^{n} Y_k^T Y_k \right) Z_h^T \\ &= Z_h \left(A_h \right) Z_h^T \end{aligned} \qquad (3.84)$$

$$A_h = Y_h^T Y_h - m_o^2 \sum_{\substack{k=1 \\ k \neq h}}^{n} Y_k^T Y_k \qquad (3.85)$$

for $h = 1, 2, \cdots, n$ where A_h is obviously a $n \times n$ symmetric square matrix. The Lagrange multipliers (E5) can now be applied to maximize the value of d'_h here with the constraint that $Z_h Z_h^T = 1$. This is done by defining a function f'_h to be

$$\begin{aligned} f'_h &= d'_h + \lambda_h (1 - Z_h Z_h^T) \\ &= Z_h (A_h) Z_h^T + \lambda_h (1 - Z_h Z_h^T) \end{aligned} \qquad (3.86)$$

where λ_h is a constant value. Differentiating f'_h with respect to Z_h

and setting the resulting expression to zero gives

$$Z_h \, A_h = \lambda_h \, Z_h \tag{3.87}$$

where λ_h and Z_h are obviously the eigenvalue and eigenvector of the square matrix A_h. The corresponding value of d_h' now becomes

$$\begin{aligned} d_h' &= Z_h \, (A_h) \, Z_h^T \\ &= Z_h \, \lambda_h \, Z_h^T \\ &= \lambda_h \end{aligned} \tag{3.88}$$

which is the eigenvalue of the matrix A_h. Since there are altogether n sets of eigenvalues and eigenvectors for the $n \times n$ square matrix A_h, the vector Z_h that maximizes the function d_h' is the eigenvector of A_h associated with the largest eigenvalue. Furthermore, it can be seen that eqn. 3.87 that there are two eigenvectors Z_h and $-Z_h$ associated with an eigenvalue of A_h. The vector Z_h to be used here is selected to be the eigenvector that gives

$$Y_h Z_h^T = \left| Y_h Z_h^Y \right| > 0 \tag{3.89}$$

Thus, the n-component vector Z_h to be used in the iterative process (Fig. 3.14) of any of the systems 9 - 11 is selected as the eigenvector of A_h (eqn. 3.85) associated with the largest eigenvalue and with the constraints that $\| Z_h \| = 1$, $Y_h Z_h^T = \left| Y_h Z_h^T \right|$, where Y_h is the n-component vector defined by eqn. 3.9. This is carried out for $h = 1, 2, \cdots, n$, so that all the n n-component vectors $\{Z_h\}$ to be used here are derived according to the method just described.

The constant m_o^2 to be used in eqn. 3.83 for the selection of Z_h is now discussed. Ideally, the value of m_o^2 should be such that when the value of d_h' (eqn. 3.83) is maximized, the value of d_h (eqn. 3.82) is also maximized. Although, in general, it does not seem to be possible to have such a value of m_o^2, it appears (by inspecting eqns. 3.82 and 3.83) that the optimum value of m_o should have the same order of magnitude as that of the value of m_k $(1 \leq k \leq n)$. Since the value of m_k, being equal to $\pm (s_k - x_k)$, can be any of the values ± 2, ± 1, 0, the value of m_o considered in this thesis is confined to be between 1 to 3.

Having described the selections of the n n-component vectors $\{Z_h\}$ to be used in the iterative process (Fig. 3.14) here, the operation of each of the systems 9 - 11 is now described as below.

System 9

System 9 is a simple modification of system 8.1 and it uses the iterative process of Fig. 3.14 where the networks $\{Z_h^T\}$ here are derived by using the Lagrange multipliers described previously. Thus, this system operates by first setting the initial values of the n components $x_1, x_2, \cdots, x_n$ of the vector X (eqn. 3.13) to zero. The initial vector of R_a (eqn. 3.22) is next set to the corresponding vector of $R' - XY$, where R' is the n-component vector defined by eqn. 3.7 and Y is the $n \times n$ matrix defined by eqn. 3.9. Having set the n components of X and R_a to their respective initial values, the iterative process of system 9 now operates as follow. The value of e_1 (eqn. 3.72) is first evaluated at the output of the network Z_1^T shown in Fig. 3.14. The value of Δx_1 is then determined from this value of e_1 according to the decision

rule given in Tables 3.1 and 3.8. The value of x_1 and the vector R_a are then updated by using eqns. 3.30 and 3.31 respectively. Having updated the value of x_1 and the vector R_a, the value of e_2 is next evaluated at the output of the network Z_2^T shown in Fig. 3.14. The value of Δx_2 is then determined according to Tables 3.1 and 3.8. The value of x_2 and the vector R_a are then updated by using eqns. 3.30 and 3.31 respectively. The iterative process continues to operate in this way with the sequential adjustment (updating) of x_3 to x_n, using the arrangement shown in Fig. 3.14. This completes the first iterative cycle of the iterative process here. Each subsequent iterative cycle operates in the same way as for the first iterative cycle except that Table 3.2 is now used in place of Table 3.1 to determine the value of Δx_h $(1 \leq h \leq n)$ in the updating process for x_h. In the last iterative cycle, however, only the updating process for x_1 is carried out and the value of x_1 obtained here is then taken as the detected value of the data symbol s_1.

In the iterative process of system 9 just described, a counter is used to count the number of iterative cycles and when it exceeds a preset threshold value, the iterative process is switched to its last iterative cycle before it is terminated.

If a sequential operation is taken to include all the operations involved in the updating process for a component of X, then it can be seen from the descriptions given above that the total number of sequential operations involved in the detection process for s_1 in system 9 is

$$n_s = (n)(n_c - 1) + 1 \qquad (3.90)$$

where n_c is the number of iterative cycles used in the iterative process here.

System 10

System 10 is a modification of system 9, in that it uses two separate but similar iterative processes (instead of one in system 9) in the detection of the data symbol s_1 from R' (eqns. 3.5 - 3.9). Each of these two iterative processes employs the same piece of equipment shown in Fig. 3.14 where the networks $\{Z_h^T\}$ here are derived by using the Lagrange multipliers described previously. Thus, in the first iterative process here, the first component x_1 of the n-component vector X (eqn. 3.13) is set to 1 and held at this value during the whole of the iterative process. The initial vectors of X and R_a (eqn. 3.22) here are as given by eqns. 3.33 and 3.34 respectively. The iterative process of Fig. 3.14 is then applied here to the n-1 components $x_2, x_3, \cdots, x_n$ of X, starting from the updating process for x_2 to that for x_n for each iterative cycle. The updating process for x_h $(2 \leq h \leq n)$ here includes the evaluation of the quantity e_h (eqn. 3.72) at the output of the network Z_h^T (Fig. 3.14), the determination of the value of Δx_h by using the appropriate threshold device (Tables 3.1 and 3.8 for the first iterative cycle, and Tables 3.2 and 3.8 for the subsequent iterative cycles), and the updating of x_h and R_a by using eqns. 3.30 and 3.31 respectively. At the end of the iterative process (when the number of iterative cycles exceeds the preset threshold value), the Euclidean distance between the vector R' and the vector XY is measured as the quantity $\|R' - XY\|$. The whole process just described is then repeated for the second iterative process but now x_1 is set and held at -1. The initial vectors of X and R_a here are now given by eqns. 3.35 and 3.36 respectively. The second iterative process is otherwise the same as the first iterative process, and at the end of this second iterative process, the quantity $\|R' - XY\|$ is evaluated. The detected value of s_1 is now taken to be the value of x_1 associated with the smaller of the two distances $\{\|R' - XY\|\}$. The total number of sequential operations n_s involved in the detection of s_1 here is obviously the same as that given by eqn. 3.54.

System 11

System 11 is a modification of system 9, in that it uses the values of the components of X (eqn. 3.13) obtained at the end of a detection process as the initial values of the corresponding components of **X** for the next detection process. Thus, the initial vector of X here is set to that given by eqn. 3.37, and the initial vector of R_a (eqn. 3.22) is set to the corresponding vector of R' - XY. The iterative process of Fig. 3.14 is then applied here to the n components $x_1, x_2, \cdots, x_n$ of X, starting from the updating process for x_1 to that for x_n for each iterative cycle. The updating process for x_h $(1 \leq h \leq n)$ here includes the evaluation of e_h (eqn. 3.72) at the output of the network Z_h^T (Fig. 3.14), the determination of the value of Δx_h by using the appropriate threshold device (Table 3.2), and the updating of x_h and R_a by using eqns. 3.30 and 3.31 respectively. The threshold values $\{t_h\}$ to be used in Table 3.2 to determine the corresponding values of $\{\Delta x_h\}$ are here given as

$$\begin{aligned} t_h &= |Y_h Z_h^T| \quad , \quad \text{if} \quad x_h = \pm 1 \\ &= 0 \quad , \quad \text{if} \quad x_h = 0 \end{aligned} \tag{3.91}$$

for $h = 1, 2, \cdots, n$. At the end of the iterative process (when the number of iterative cycles exceeds the preset threshold value), the vector X is taken as the detected data-symbol vector S' of S, so that $s_1' = x_1, s_2' = x_2, \cdots, s_n' = x_n$ where $s_1', s_2', \cdots, s_n'$ are the n components of S'. However, only the value of s_1' is here accepted as the detected value of the data symbol s_1. The last n-1 components of S' are then shifted one place to the front to become the first n-1 components of the initial vector of X for the next detection process.

It can be seen from the descriptions given above that the total number of sequential operations n_s involved in the detection of s_1 in system 11 is the same as that given by eqn. 3.47.

It is shown in appendix A5 that the iterative process (Fig. 3.14) of any of the systems 9 - 11 is not prevented from diverging, so long as the networks $\{Z_h^T\}$ used here are not the same as the networks $\{Y_h^T\}$ used in Fig. 3.4. Divergence is here defined to have occurred when the Euclidean distance $\|R' - XY\|$ increases during the iterative process, bearing in mind that it is desirable here to select, as the detected vector of S, the vector of X that has the minimum value of $\|R' - XY\|$ (section 3.3). It is also shown in appendix A7 that the sufficient condition for error-free detection to be achieved in the absence of noise is

$$|Y_1 Z_1^T| > 2 \sum_{k=2}^{n} |Y_k Z_1^T| \qquad (3.92)$$

for any of the systems 9 and 11, and is

$$|Y_h Z_h^T| > 2 \sum_{k=h+1}^{n} |Y_k Z_h^T| \qquad (3.93)$$

for $h = 2, 3, \cdots, n$ for system 10. Thus, after evaluating the $\{Z_h\}$ for the iterative process (Fig. 3.14) of any of the systems 9 - 11, the corresponding inequality of (3.92) or (3.93) should always be checked to ensure that the system is able to operate correctly at least in the absence of noise.

The tolerances to additive white Gaussian noise of systems 9 - 11 operating over channel A (Table 2.1) have been studied by using computer simulation tests and the results are shown in Figs. 3.17 - 3.21. The 95% confidence limits of the results here are about $\pm$0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and 3.40 respectively. The optimum system here is, of course, the system described in section 3.3.

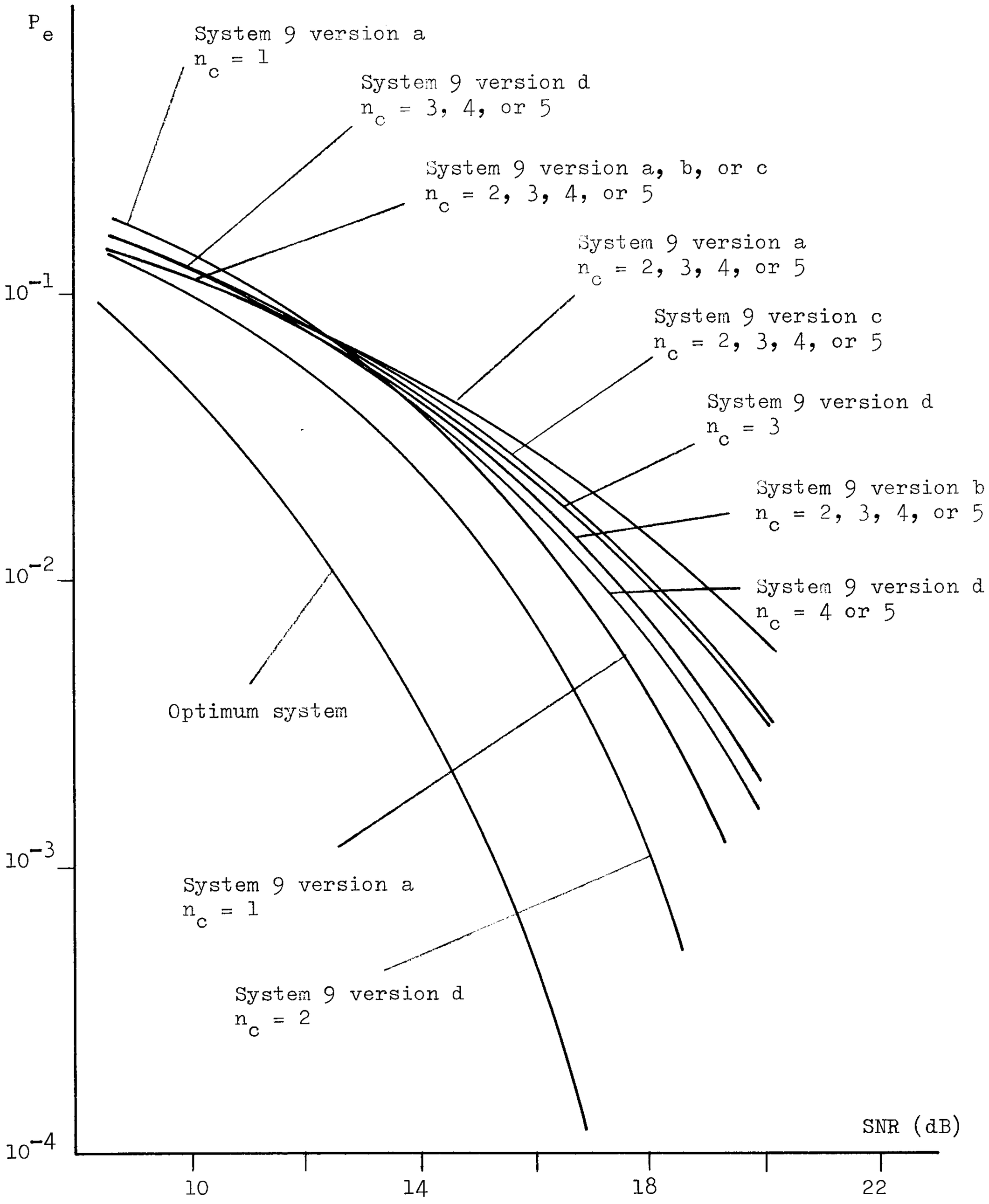


Fig. 3.17 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 9 with versions a - d (Table 3.8) operating over channel A (Table 2.1).
m_o = 1 (eqn. 3.83).
n_c : number of iterative cycles.

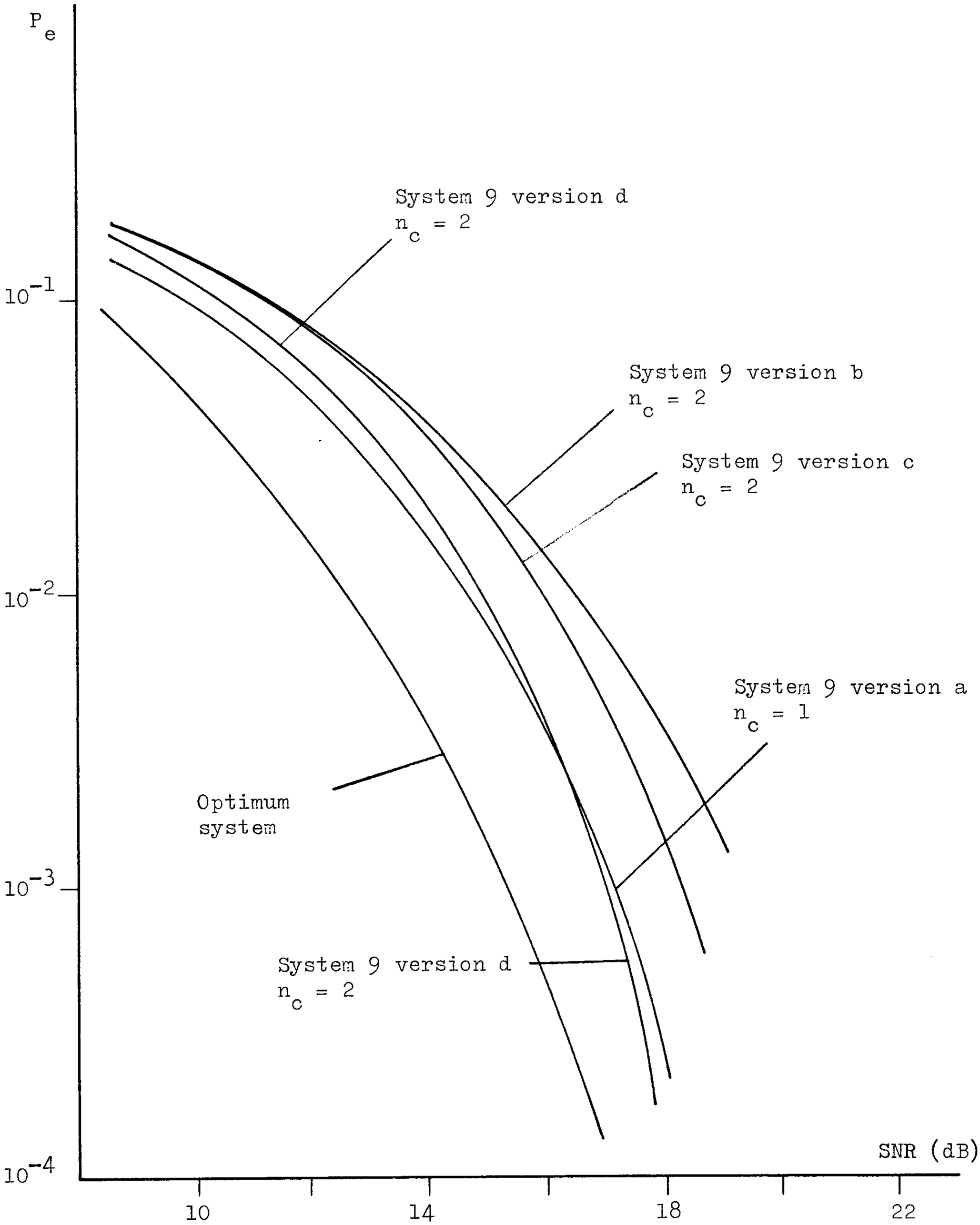


Fig. 3.18 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 9 with versions a - d (Table 3.8) operating over channel A (Table 2.1).
$m_o = 2$ (eqn. 3.83).
n_c : number of iterative cycles.

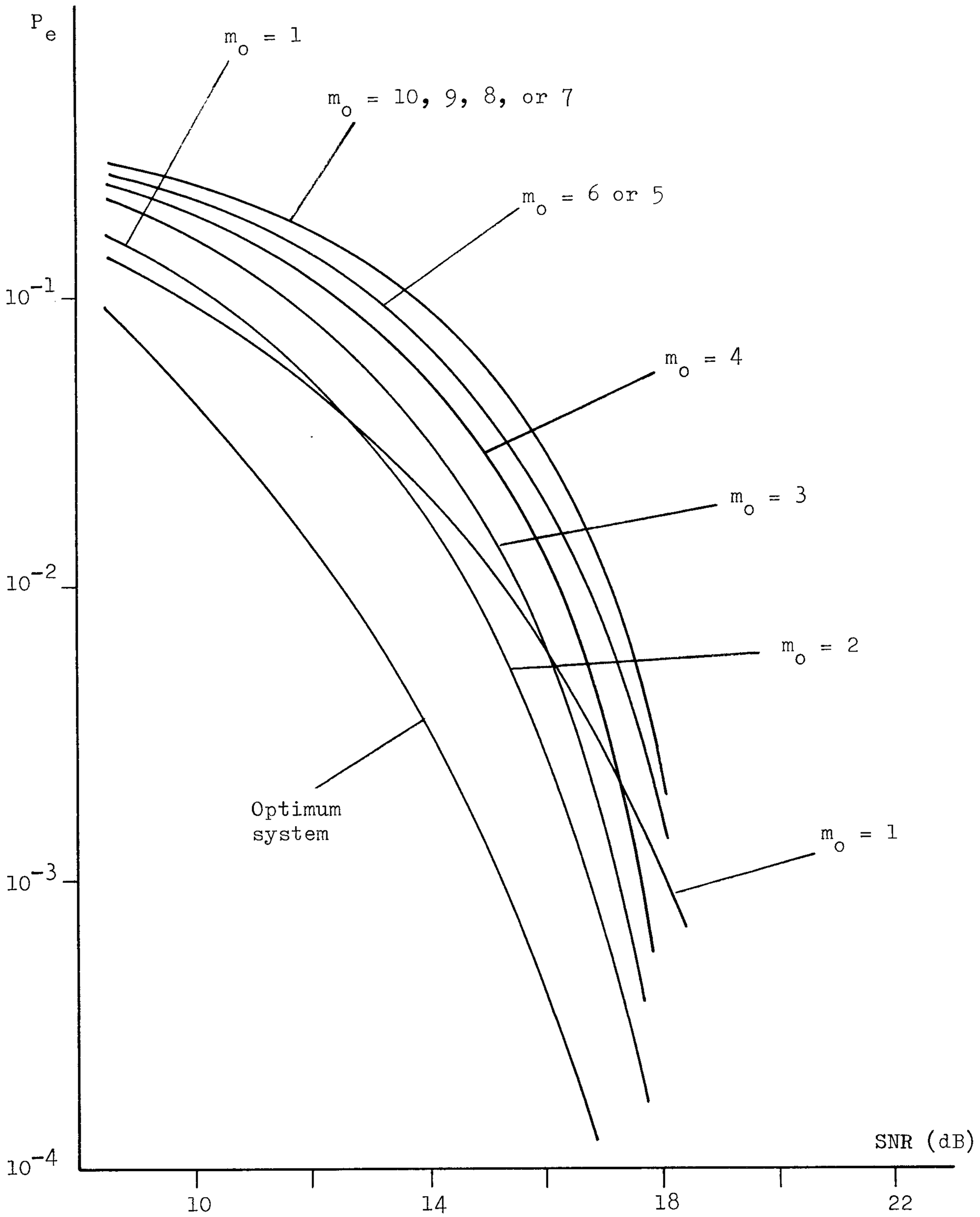


Fig. 3.19 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 9 with version d (Table 3.8) operating over channel A (Table 2.1).
Number of iterative cycles $n_c = 2$.
m_o : constant defined in eqn. 3.83.

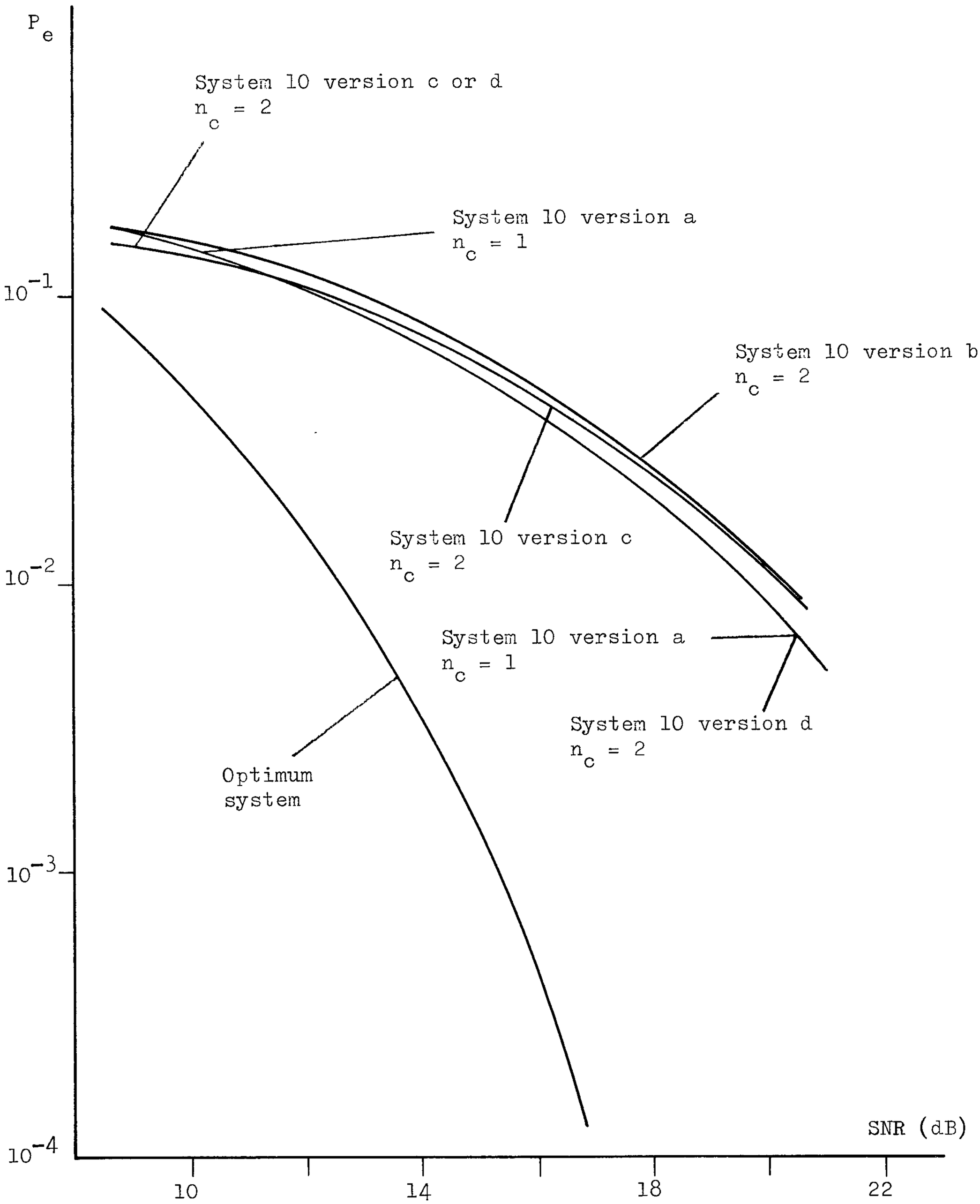


Fig. 3.20 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 10 with versions a - d (Table 3.8) operating over channel A (Table 2.1).
m_o = 1 (eqn. 3.83).
n_c : number of iterative cycles.

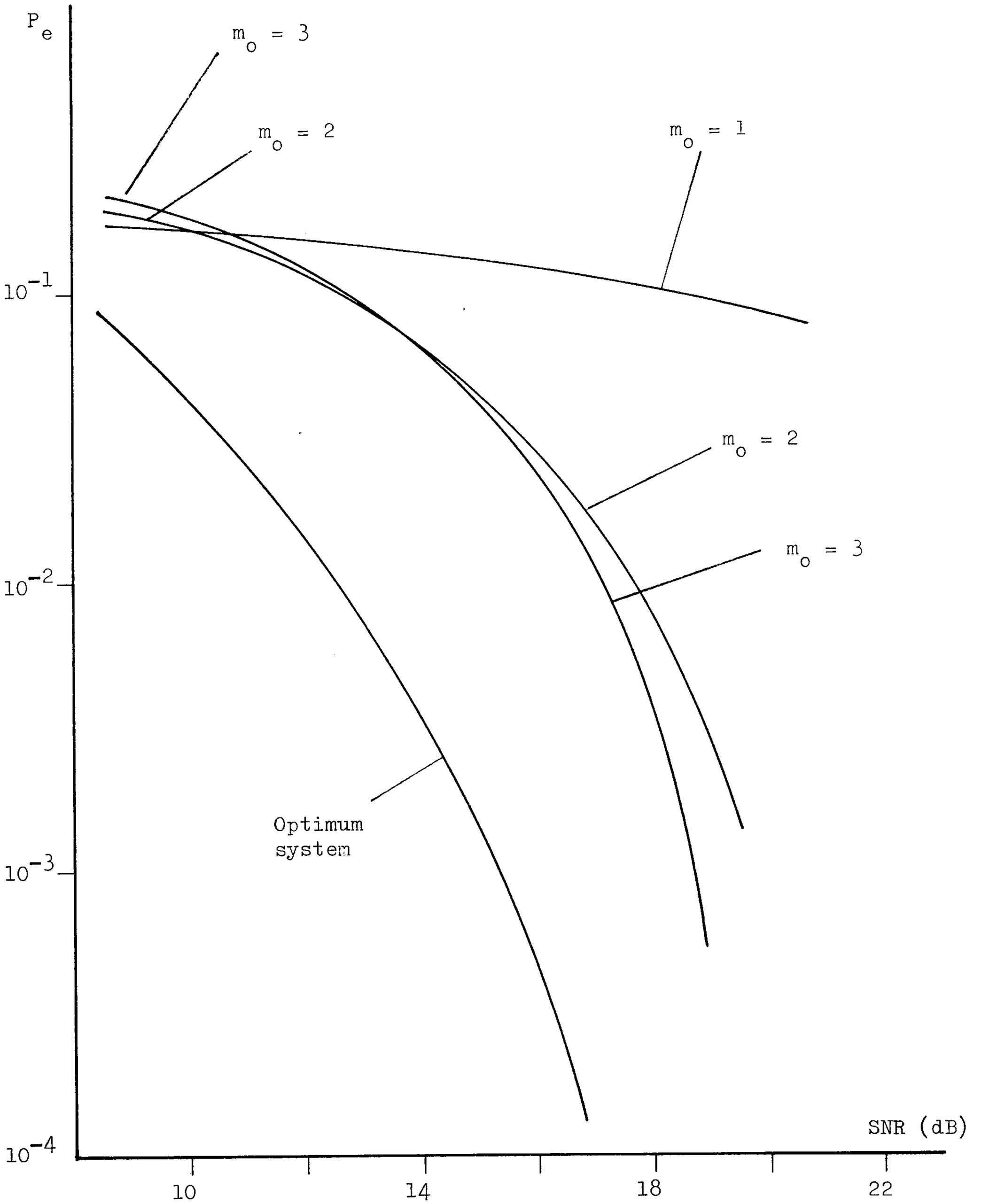


Fig. 3.21 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 11 with version d (Table 3.8) operating over Channel A (Table 2.1).
Number of iterative cycles $n_c = 2$.
m_o : constant defined in eqn. 3.83.

Fig. 3.17 shows the tolerances to noise of the various versions (Table 3.8) of system 9 with $m_o = 1$, where m_o is the constant defined in eqn. 3.83. As it appears, the performance of system 9 can sometimes deteriorate when the number of iterative cycles n_c used in the iterative process here increases. This suggests that divergence has occurred during the iterative process of system 9, and the number of iterative cycles must now be chosen carefully to achieve the best tolerance to noise. It can be seen from Fig. 3.17 that the best tolerance to noise is achieved when $n_c = 1$ for system 9 with version a, or when $n_c = 2$ for system 9 with any of the versions b, c, and d. Furthermore, it can be seen from Tables 3.1, 3.2, and 3.8 that x_1 may not have a possible value of s_1 (where $s_1 = 1$, or -1) unless $n_c \geq 1$ for system 9 with version a, or $n_c \geq 2$ for system 9 with any of the versions b, c, and d. It therefore appears that in order to achieve the best tolerance to noise for system 9 with any of the versions a, b, c, and d, the iterative process here should use the minimum number of iterative cycles that ensures x_1 to have a possible value of s_1. Among all the various versions of system 9 with $m_o = 1$ considered in Fig. 3.17, version d with $n_c = 2$ appears to have the best tolerance to noise here.

Fig. 3.18 shows the tolerances to noise of the various versions (Table 3.8) of system 9 with $m_o = 2$. Again, the best tolerance to noise of system 9 appears to be achieved when version d with $n_c = 2$ is used in the iterative process here. This arrangement of system 9 (that is, version d with $n_c = 2$) is now selected to investigate the effect of the value of m_o on the system's performance, and the results are shown in Fig. 3.19.

As Fig. 3.19 shows, system 9 with version d and $n_c = 2$ appears to perform better as the value of m_o decreases from 10 to 2. It is also observed that all the curves of system 9 with $m_o = 2, 3, \cdots, 10$ tend

to converge to give the same tolerance to noise at high signal to noise ratios when the error rates fall below 10^{-4}. The arrangement of $m_o = 1$, however, only appears to perform well at low signal to noise ratios, and its performance becomes inferior as compared to the other arrangements here at high signal to noise ratios. System 9 with version d and $n_c = 2$ therefore appears to have the best tolerance to noise here when $m_o = 2$.

It can be seen from Fig. 3.19 that, the tolerance to noise of system 9 with version d, $n_c = 2$, and $m_o = 2$ is about 1.5 dB inferior to that of the optimum system at an error rate of 10^{-3}. The number of sequential operations involved in the detection process for s_1 in system 9 here is 9 (eqn. 3.90) which is a very small number. System 9 therefore appears to be very promising when operating over channel A (Table 2.1).

Fig. 3.20 shows the tolerances to noise of the various versions (Table 3.8) of system 10 with $m_o = 1$ operating over channel A (Table 2.1). It can be seen from Figs. 3.17 and 3.20 that the tolerance to noise of system 10 is inferior to that of system 9 here. This is because the iterative process used in any of the systems here is not prevented from diverging, so that for a given number of iterative cycles used in the iterative process, the distance $\|R' - XY\|$ obtained at the end of the detection process for s_1 is now likely to be smaller in system 9 than in system 10, bearing in mind that only the first component of X is updated in the last iterative cycle in system 9 whereas all the n components of X are updated in the last iterative cycle in system 10. It should perhaps be reminded that the optimum performance is achieved when the vector X (being a possible vector of S) obtained at the end of the detection process for s_1 has the smallest value of $\|R' - XY\|$ (section 3.3). The same reason here can be used to explain the inferior performance of system 11 in relation to that of system 9 (Figs. 3.17 and 3.21).

3.14 Further Computer Simulation Results

All the systems described in this chapter have so far been tested only on channel A (Table 2.1) which introduces severe pure amplitude distortion. The tolerances to additive white Gaussian noise of the more important systems here are shown again in Fig. 3.22. In order to have a better understanding of these more important systems, further computer simulation tests are carried out to determine the tolerances to noise of these systems operating over channels B and C whose corresponding sampled impulse responses are as given in Table 2.1. These two channels are selected to have different characteristics. Thus, channel B here introduces a combination of both phase and amplitude distortions, whereas channel C here introduces very severe amplitude distortion but no phase distortion. The number of components in the sampled impulse response of channel C is also larger than the number of samples, n, used in the iterative process of each of the various systems here (where n = 8). In these tests, 10,000 data symbols $\{s_i\}$ are transmitted for each measurement of the bit error rate, and the results are shown in Figs. 3.23 and 3.24. The definitions of the bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and 3.40 respectively. When the simulation results appear to be more scattered which normally occur at low error rates ($< 10^{-3}$), more simulation tests are carried out to measure the bit error rates at the same signal to noise ratios. The 95% confidence limits of the results shown in Figs. 3.22 - 3.24 are about ± 0.5 dB. The optimum system here is, of course, the system described in section 3.3.

All computer simulation tests have been carried out by using the Prime 400 computer at the Loughborough University of Technology. The computer programs here are written in FORTRAN. The computer program for system 5 is also shown in appendix B1.

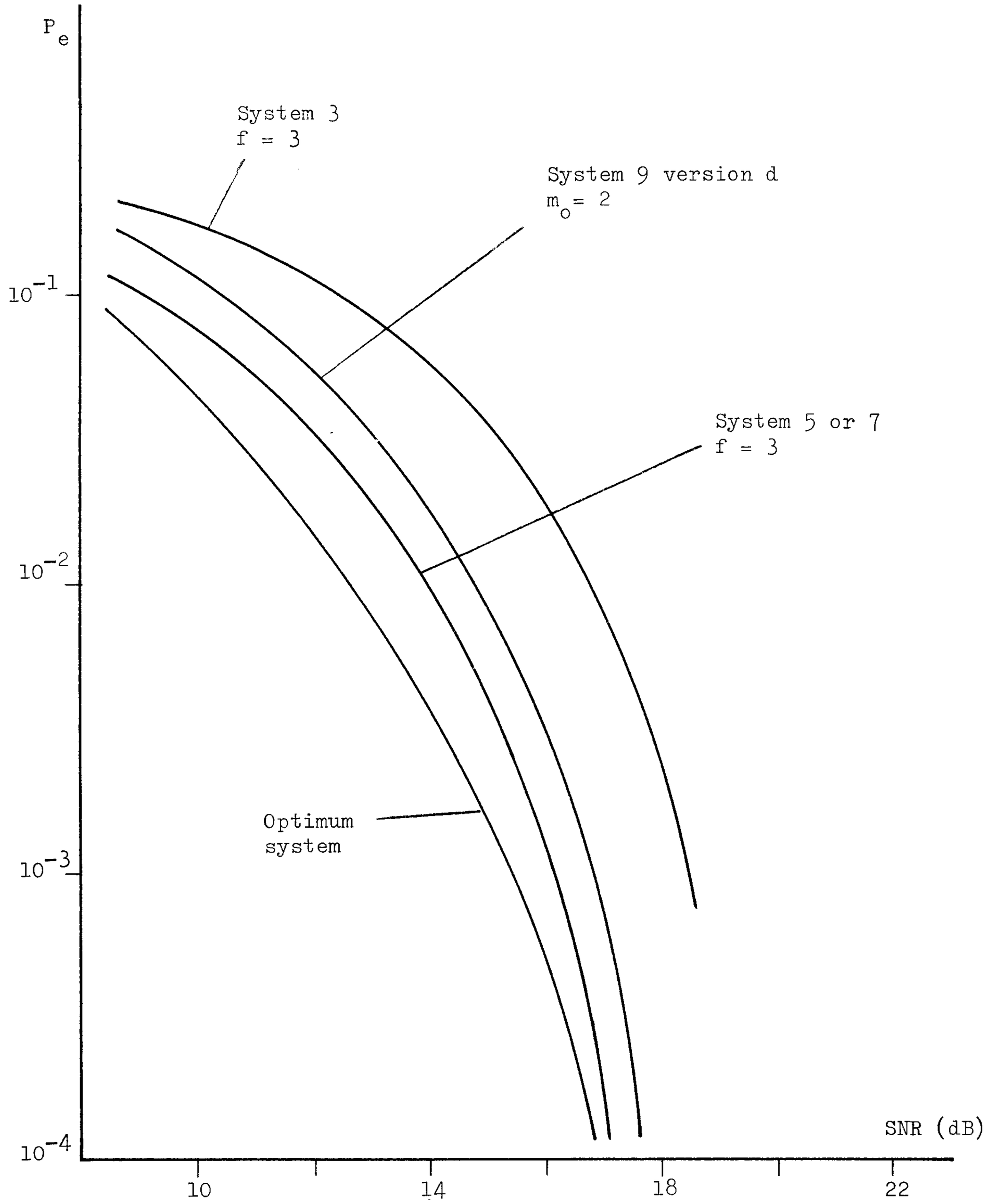


Fig. 3.22 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for the more promising systems operating over channel A (Table 2.1).
Number of iterative cycles $n_c = 2$.
f : integer defined in eqn. 3.41 or in eqns. 3.58 and 3.59.
m_o : constant defined in eqn. 3.83.

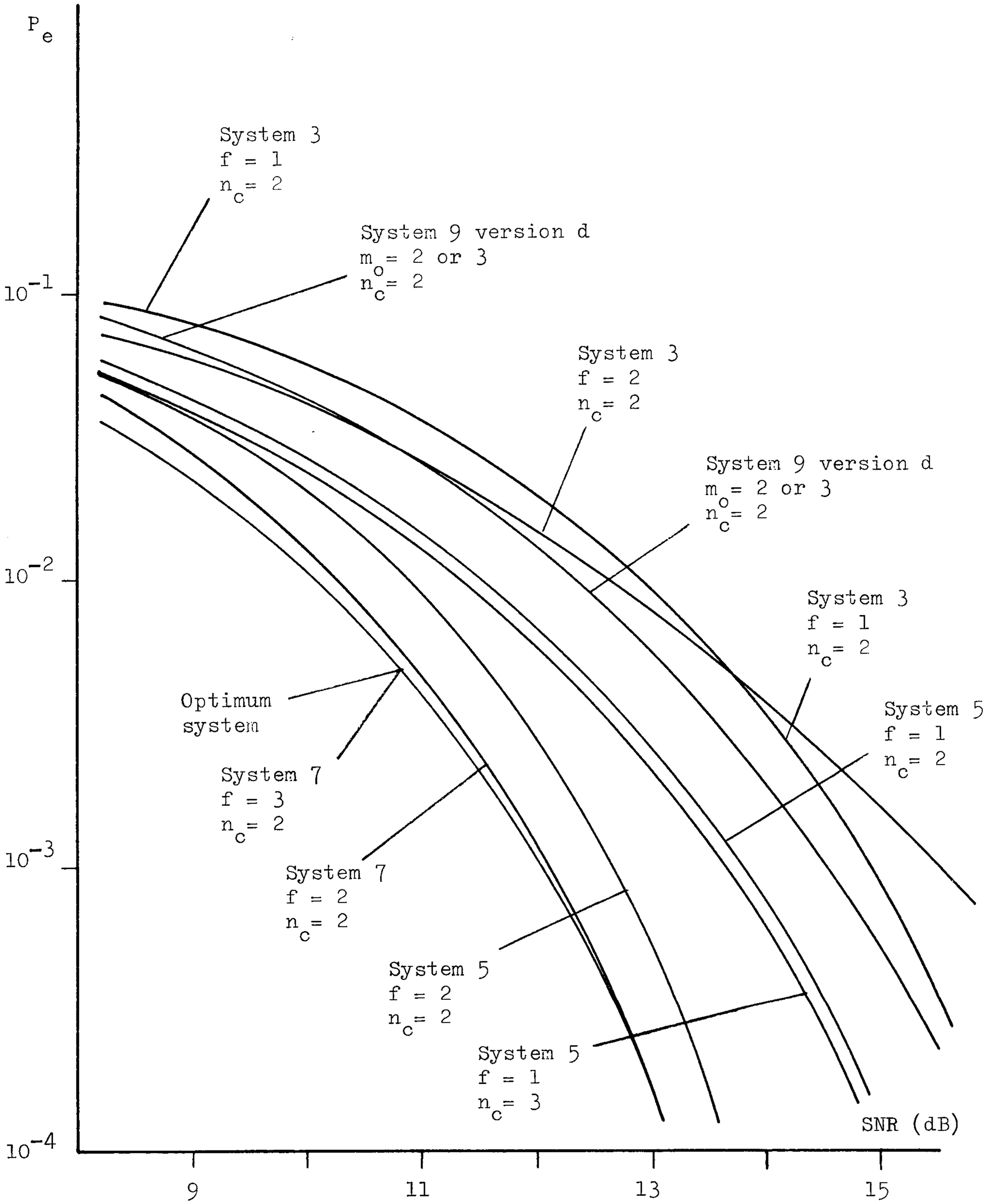


Fig. 3.23 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for the more promising systems operating over channel B (Table 2.1).

n_c : number of iterative cycles.

f : integer defined in eqn. 3.41 or in eqns. 3.58 and 3.59.

m_o : constant defined in eqn. 3.83.

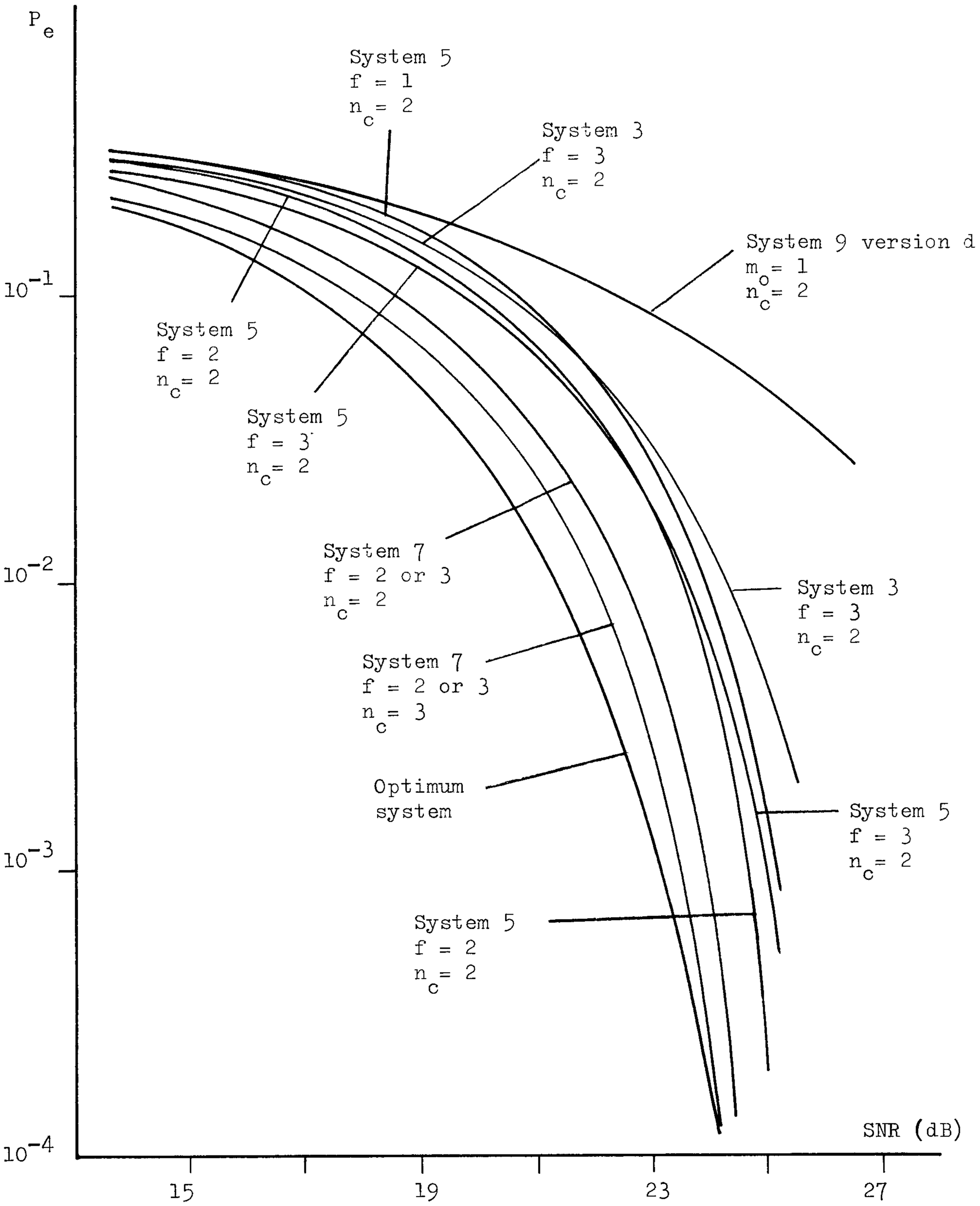


Fig. 3.24 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for the more promising systems operating over channel C (Table 2.1).

n_c : number of iterative cycles.

f : integer defined in eqn. 3.41 or in eqns. 3.58 and 3.59.

m_o : constant defined in eqn. 3.83.

3.15 Assessment of Systems

The operations of the more important systems are summarised as below.

System 3 : The iterative process here operates from x_{n-f+1} to x_n for the first iterative cycle, and then from x_1 to x_n for each subsequent iterative cycle, using the arrangement of Fig. 3.4. The initial vector of X here is

$$X_o = [\ s'_2 \quad s'_3 \quad \cdots \quad s'_{n-f+1} \quad \overbrace{0 \quad \cdots \quad 0}^{f}\ ]$$

where $s'_2, s'_3, \cdots, s'_{n-f+1}$ are the detected values of the corresponding data symbols obtained at the end of the previous detection process for s_1, and f is a small integer. The detected value of s_1 is taken as the value of x_1 obtained at the end of the iterative process.

System 5 : Two separate but similar iterative processes are used here. Each of these two iterative processes operates in the same way as for the iterative process of system 3. The initial vectors of X for the two iterative processes here are

$$(X_o)_1 = [\ s'_2 \quad s'_3 \quad \cdots \quad s'_{n-f} \quad s'_{n-f+1} \quad \overbrace{0 \quad \cdots \quad 0}^{f}\ ]$$

$$(X_o)_2 = [\ s'_2 \quad s'_3 \quad \cdots \quad s'_{n-f} \quad -s'_{n-f+1} \quad \overbrace{0 \quad \cdots \quad 0}^{f}\ ]$$

for $0 < f < g$. The detected value of s_1 is taken as the value of x_1 obtained at the end of the iterative process associated with the smaller value of $\|R' - XY\|$.

System 7 : Four separate but similar iterative processes are used here. Each iterative process here operates in the same way as for the iterative process of system 3 except that each of the last two iterative processes here involves only the updating process for x_n for the first iterative cycle. The initial vectors of X used in the first two iterative processes are the same as those used in system 5. The initial vectors of X for the last two iterative processes here are

$$(X_o)_3 = \begin{bmatrix} s'_2 & s'_3 & \cdots & s'_n & 0 \end{bmatrix}$$

$$(X_o)_4 = \begin{bmatrix} -s'_2 & -s'_3 & \cdots & -s'_n & 0 \end{bmatrix}$$

The detected value of s_1 is taken as the value of x_1 obtained at the end of the iterative process associated with the smallest value of $\|R' - XY\|$.

System 9 : The iterative process here operates from x_1 to x_n for each iterative cycle, except the last iterative cycle where only the value of x_1 is updated, using the arrangement of Fig. 3.14. The networks $\{Z_h^T\}$ shown in Fig. 3.14 are here selected to maximize the corresponding quantities $\{d'_h\}$ defined by eqn. 3.83, so as to hopefully maximize the relative magnitude of the signal component and the intersymbol interference component in the detection of that signal, for a given noise level.

The initial vector of X here is

$$X_o = \begin{bmatrix} \overbrace{0 \quad \cdots \quad 0}^{n} \end{bmatrix}$$

The detected value of s_1 is taken as the value of x_1 obtained at the end of the iterative process.

The number of sequential operations n_s involved in the detection process of s_1 for each of the systems shown in Figs. 3.22 - 3.24 is summarised in Table 3.11. A sequential operation here is taken to include all the operations involved in the updating process for one component of the n-component vector X during the iterative process (Table 3.3). The number of distance measurements required in each system here is also shown in Table 3.11, where the distance is defined as the quantity $\|R' - XY\|$. In practice, the value of $\|R' - XY\|^2$ will be evaluated instead of $\|R' - XY\|$. Thus, a distance measurement here involves squaring the n components of the vector R' - XY, and adding the resultant values. The amount of operations involved here can be seen to be about the same as that involved in the updating process for one component of X in the iterative process that employs the arrangement of Fig. 3.4 or 3.14 (Table 3.3). The amount of operations involved in a distance measurement here may therefore be considered as roughly the same as that involved in a sequential operation shown in Table 3.11.

It can be seen from Fig. 3.24 that system 9 has a very poor tolerance to noise over channel C which is known to introduce very severe amplitude distortion. Figs. 3.22 - 3.24 also show that systems 3 and 9 do not seem to perform as well as systems 5 and 7 over any of the channels A, B, and C. Nevertheless, systems 3 and 9 are important in the sense that very few sequential operations and no distance measurement are required in the detection process for s_1 in these systems, as can be seen from Table 3.11. These systems are therefore able to operate with very high speed.

System 5 is probably the most promising system here in terms of the error rate performance and the amount of operations involved in the system. For each of the three channels tested here, system 5 requires about 22 sequential operations to achieve a performance near to that of the optimum

Systems	Number of Iterative Cycles n_c	Number of Sequential Operations n_s	Number of Distance Measurements n_d
3 (f=1)	2	9	-
3 (f=2)	2	10	-
3 (f=3)	2	11	-
5 (f=1)	2	18	2
5 (f=1)	3	34	2
5 (f=2)	2	20	2
5 (f=3)	2	22	2
7 (f=2)	2	38	4
7 (f=2)	3	70	4
7 (f=3)	2	40	4
7 (f=3)	3	72	4
9	2	9	-
Optimum system	-	-	256

Table 3.11 Number of sequential operations and distance measurements required in the more important systems of this chapter (Figs. 3.22 - 3.24). f is the appropriate integer used in the initial vector of X for the iterative process. Distance is here referred to $\|R' - XY\|$.

system. In particular, at an error rate of 10^{-3}, the tolerance to noise of system 5 is only about 0.5 dB lower than that of the optimum system when operating over channel B. For channels A and C, system 5 is about 1 dB and 1.5 dB, respectively, inferior to the optimum system at an error rate of 10^{-3}. In any case, the difference in tolerance to noise between system 5 and the optimum system decreases as the error rate decreases below 10^{-3}.

System 7 has a performance approaching that of the optimum system especially when it is operating over channel B or C. However, the number of sequential operations and distance measurements involved here is often about twice of that involved in system 5.

The weakness of system 9 is that divergence tends to occur during the iterative process here so that the number of iterative cycles used must always be kept to the minimum to achieve the best performance. The powerful iterative technique of gradually eliminating the intersymbol interference components in e_1 (eqn. 3.72) in determining the detected value of s_1 is therefore lost in this system. Furthermore, the networks $\{Z_h^T\}$ of Fig. 3.14 used here are not easily evaluated from the channel sampled impulse response. This means that system 9 can only be used in the data-transmission system operating over a known time-invariant channel, where the $\{Z_h^T\}$ can always be evaluated before the transmission of data begins.

In any of the systems 3, 5, and 7, the iterative process is always prevented from diverging after the first iterative cycle, so that the quantity $\|R' - XY\|$ does not increase in the subsequent operations. If the vector X obtained at the end of the iterative process here is always the same as the possible vector of S that gives the minimum value of $\|R' - SY\|$, then the performance of the optimum system described in section 3.3 is

achieved here. However, as is explained in section 3.7, the iterative process here effectively operates by changing one component of S at a time (to its other possible value) to search for a smaller value of $\|R' - SY\|$, and it may therefore cease to proceed further before the minimum value of $\|R' - SY\|$ is reached. It therefore appears to be important to begin the iterative process here with an initial vector of X that has a value of $\|R' - XY\|$ close to the minimum value of $\|R' - SY\|$. At high signal to noise ratios, this can be done by using the detected values of some or all components of S obtained at the end of a detection process as the initial values of the corresponding components of X for the next detection process. This is the basis of the iterative process used in the detection process for system 3. By using an additional but similar iterative process which assumes that one of the last few components of S has been detected with error in the previous detection process, system 5 is able to achieve a near-optimum performance with as few as 22 sequential operations and 2 distance measurements. The optimum system here, however, requires 256 distance measurements, bearing in mind that the amount of operations involved in a distance measurement is roughly the same as that involved in a sequential operation in the iterative process of system 5. The tolerance to noise of system 5 can of course be expected to improve yet further by using more similar iterative processes in the detection process, with a trade off in equipment complexity and amount of operations involved in the system, as can be seen from the development of system 7. Apart from being able to give very satisfactory tolerance to noise, each of the systems 3, 5, and 7 is probably also well suited for use over a time varying channel. This is because all the parameters required in the detection process here, such as the $\{Y_h^T\}$ in Fig. 3.4, can be evaluated easily from the channel sampled impuse response which in turn can be estimated relatively simply and rapidly by using a channel estimator.(A9, C39)

To conclude, a very promising iterative detection process implemented as system 5 has been developed in this chapter. It has the advantage of being able to operate with a very small amount of operations and hence a very high speed, while having a near-optimum tolerance to noise. The equipment involved in this system is also very simple enabling it to be used over a time varying channel. Furthermore, a different type of systems that uses a different set of networks in place of the $\left\{Y_h^T\right\}$ used in the Gauss-Seidel iterative process has been studied and found to give an inferior performance. Finally, the properties and the behaviours of the iterative detection processes based on the Gauss-Seidel iterative process have been studied and analysed and a much better understanding of this type of detection process has been achieved.

CHAPTER 4

SYSTEMATIC SEARCH DETECTION PROCESS FOR BINARY BASEBAND SIGNALS

4.1 Introduction

In chapter 3, an iterative process is used to search for the received signal vector in the detection process. This chapter presents a different approach to the problem. Basically, the new technique here involves partitioning systematically the vector space that contains all the possible vectors of the received signal vector in searching for the received signal vector. The object here is to develop a system that can operate with a very small amount of some simple operations in achieving a high tolerance to additive white Gaussian noise.

4.2 Basic Assumptions

The basic model of the data-transmission system here is the synchronous serial binary baseband data-transmission system described in section 2.1. The signal processor in Fig. 2.1 is here implemented as a detection process that employs the arrangement of detection and intersymbol interference cancellation shown in Fig. 4.1. This arrangement of the detection process is the same as that used in the systems studied in chapter 3 except that a systematic search process is now used in place of the iterative process to implement the detector here. Consequently, most of the basic assumptions described in section 3.2 are carried over to this section. Thus, the sample value at the input to the signal processor

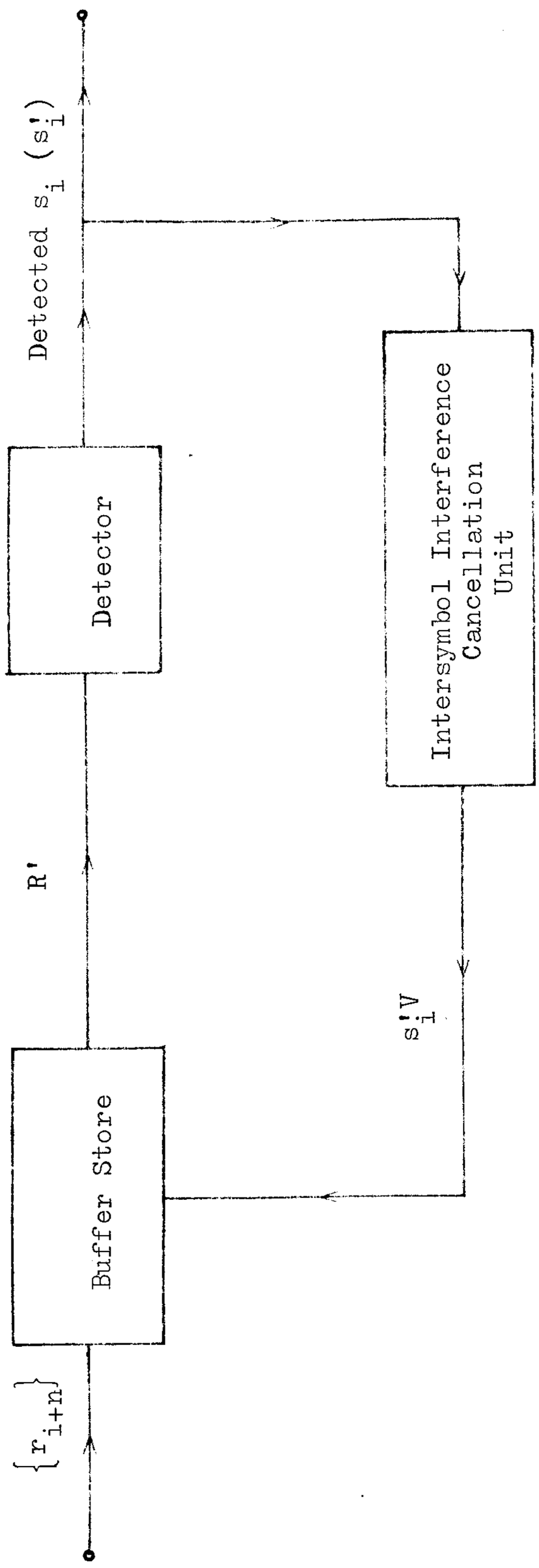


Fig. 4.1 Signal processor using the arrangement of Detection and Intersymbol Interference Cancellation

shown in Fig. 4.1, at time t = iT, is

$$r_i = \sum_{h=0}^{g} s_{i-h} y_h + w_i \qquad (4.1)$$

where the $\{s_i\}$ are the data symbols whose values are to be detected here, $y_0, y_1, \cdots, y_g$ are the g+1 components of the sampled impulse response of the baseband channel shown in Fig. 2.1, and the $\{w_i\}$ are the noise components here. The $\{s_i\}$ here are assumed to be statistically independent and equally likely to have any of the two binary values 1 and -1, and the $\{w_i\}$ are assumed to be statistically independent Gaussian random variables with zero mean and a fixed variance σ^2 where σ^2 is the two-sided power spectral density of the noise added at the output of the transmission path shown in Fig. 2.1. It is also assumed that the channel sampled impulse response is known and time-invariant, so that some pre-computation using the channel sampled impulse response can now be carried out for the detection process before the transmission of data begins.

In Fig. 4.1, the buffer store holds the n sample values of the vector R' (eqns. 3.2, 3.4, 3.7). The detector here operates on R' to give the detected value s_i' of s_i, by using a systematic search process to be described in section 4.4. The intersymbol interference cancellation unit (Fig. 4.1) then generates the (g+1)-component vector $s_i'V$ (eqn. 3.3) which is then removed (by subtraction) from R'. The resultant components of R' are then moved one place to the front to become the new R' whose last component is the next received sample value r_{i+n}. The detection process continues in this way for the detection of the next data symbol s_{i+1}.

Again, to simplify the nomenclature, rename s_i as s_1, s_{i+1} as s_2, and so on, and rename w_i as w_1, w_{i+1} as w_2, and so on. Furthermore, let S and W be the n-component vectors whose hth components are given as s_h

and w_h respectively (eqns. 3.5 and 3.6). Assuming the correct detection of the g previous data symbols, it can be seen from section 3.2 that

$$R' = SY + W \qquad (4.2)$$

where Y is the $n \times n$ matrix defined by eqn. 3.9. In eqn. 4.2, SY is the received signal vector, and W is the noise vector. Consequently, the detector in Fig. 4.1 now operates on the n-component vector R' to give the detected vector of the received signal vector. The first component of the corresponding detected vector of S is then accepted as the detected value of the data symbol s_1.

4.3 Optimum System and the Basis of the Systematic Search Process

The optimum system here is the same as the system described in section 3.3 and it operates by evaluating the distance $\|R' - SY\|$ for each of the possible vectors of S. The detected value of s_1 is then taken as the value of the first component of the possible vector of S having the minimum distance.

The weakness of this optimum system is that, in the detection of s_1 from R', all the 2^n possible vectors $\{S\}$ are considered so that a total of 2^n distance measurements are now required here which can be excessive when $n > 10$.[A9] It appears that some simple systematic tests may be carried out first on R' to select just a few of the 2^n possible vectors $\{SY\}$ which are more likely to include the received signal vector, so that only the distances of these few selected vectors are now required to be evaluated. This may result in having a much more efficient system which has approximately the same tolerance to noise as that of the optimum system while requiring far fewer operations in the detection process.

Consider first a technique that is used for the analog to digital conversion. A received real-valued quantity r which may have any value between -8 to $+8$ is required to be coded into one of the 8 discrete values -7, -5, -3, -1, 1, 3, 5, and 7. A crude and very inefficient way of doing this is to measure the absolute value $|r - q|$ of $r - q$ for each of the 8 possible q values -7, -5, -3, -1, 1, 3, 5, and 7, and to accept that value of q for which $|r - q|$ is the minimum. A much quicker method is as follows. First compare r with a threshold of 0. If $r > 0$, consider only $q = 1, 3, 5, 7$, and if $r \leq 0$, consider only $q = -1, -3, -5, -7$. Now if $r > 0$, compare r with a threshold of 4, and if $r > 4$ consider only $q = 5, 7$, and if $r \leq 4$ consider only $q = 1, 3$. Finally, if $r > 4$, compare r with a threshold of 6, and if $r > 6$ then $q = 7$, whereas if $r \leq 6$ then $q = 5$. Clearly, in the method just described, only 3 measurements are required instead of 8 in the previous crude and inefficient method. In general, where q has 2^n possible values, only n measurements are required to code r into a value of q. Furthermore, if q is equally likely to have any of its possible values, and any one q, say q_i, is statistically independent of any other, say q_j, then q carries n bits of information and n measurements are required to evaluate q from r. This appears to be the absolute minimum in the number of operations required to determine q from r.

Similarly, in the maximum likelihood detection of S from R', if the n components of S are statistically independent and are equally likely to have any of their two possible values, then S carries n bits of information, and the absolute minimum number of operations required to detect S from R' is now n instead of 2^n in the optimum system described previously. Consequently, it is now possible to obtain the detected vector of S by successive partitioning the n-dimensional Euclidean vector space

that contains R', each time reducing the number of possible vectors $\{SY\}$ (and hence $\{S\}$) to be considered to half the previous value. n decision boundaries are now required here to determine the detected vector of SY (and hence S) from R', the decision boundaries being the appropriate hyperplanes in the n-dimensional Euclidean vector space. However, the difficulty here is that, in order to eliminate from further consideration exactly one half of the number of possible vectors $\{SY\}$ at each operation such that the rejected vectors never contain that associated with the minimum distance $\|R' - SY\|$, nonlinear (crooked, with many changes in direction) decision boundaries must be used, leading to highly complex operations.

In the system now to be studied, a search process which involves simple successive partitioning of the n-dimensional Euclidean vector space containing R' is used to select a few of the 2^n possible vectors $\{SY\}$ which are more likely to include that associated with the minimum distance $\|R' - SY\|$. The detected value of s_1 is then taken as the first component of the vector S corresponding to the selected vector of SY having the minimum distance.

4.4 Detector employing the Systematic Search Process

In the detection of s_1 from R', the detector here (Fig. 4.1) first operates on R' to give the detected vector S' of S. This is carried out by using the arrangement shown in Fig. 4.2. Thus, in Fig. 4.2, the n-component vector R' is fed to the input of each of the f networks $\{Z_h^T\}$ for $h = 1, 2, \cdots, f$ where the value of f may be selected to be between 1 to n inclusive. Each network Z_h^T here may be implemented as a feedfordward transversal filter whose tap gains are given by the n

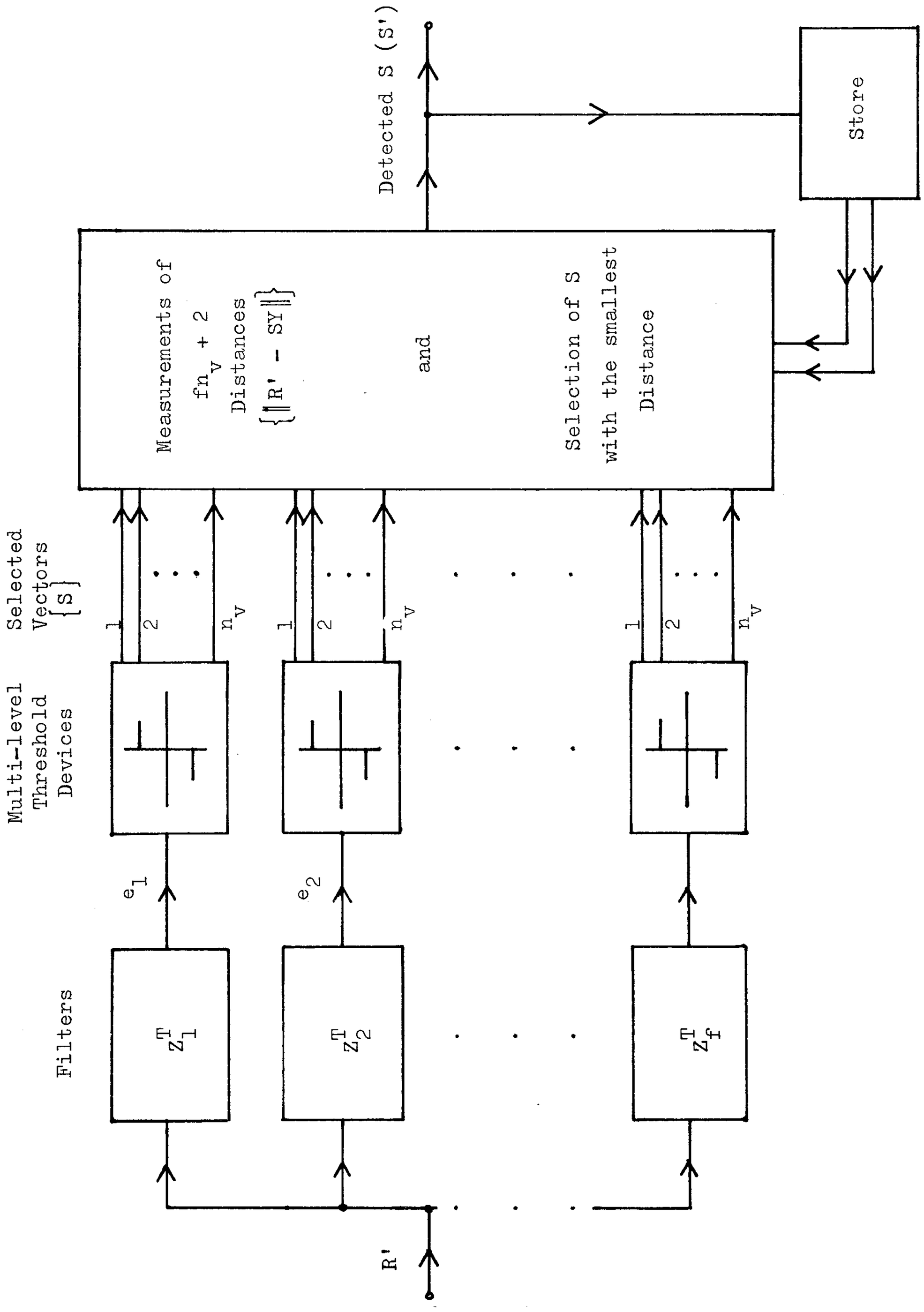


Fig. 4.2 Detection of S from R' using the systematic search process.
$1 \leq f \leq n$.

components of the n-component vector Z_h. The vectors $\{Z_h\}$ are here constrained to be unit vectors so that

$$Z_h Z_h^T = \| Z_h \|^2 = 1 \qquad (4.3)$$

for $h = 1, 2, \cdots, f$ where $\| Z_h \|$ is the Euclidean norm or length of the vector Z_h, and Z_h^T is the transpose of the row matrix Z_h, bearing in mind that a vector is also treated as a row matrix in this thesis. The quantity e_h obtained at the output of the network Z_h^T shown in Fig. 4.2 is here given by

$$e_h = R' Z_h^T \qquad (4.4)$$

for $h = 1, 2, \cdots, f$. The selections of the networks $\{Z_h^T\}$ to be used here are studied and described in sections 4.5 - 4.10. Having evaluated the values of $\{e_h\}$ from R', each of these values is fed to a multi-level threshold device shown in Fig. 4.2. n_v ($< 2^n$) possible vectors of S are then selected in each of the f threshold devices here. The structures and operations of these multi-level threshold devices will shortly be described. Thus, at the outputs of the f threshold devices, a total of fn_v possible vectors $\{S\}$ have now been selected. The distance $\|R' - SY\|$ associated each of these selected vectors is then measured and stored together with the corresponding vector S. The detector then proceeds to select another two possible vectors of S from the detected vector S' of S obtained at the end of the previous detection process. These two selected vectors are such that, the first n-1 components $s_1, s_2, \cdots, s_{n-1}$ are the same as their respective detected values obtained at the end of the previous detection process, and the last component s_n is set to 1 in one vector and -1 in the other. The distance $\|R' - SY\|$ associated with each

of these two selected vectors is then evaluated and stored together with the corresponding vector S. Thus, there are now altogether (fn_v+2) possible vectors of S being selected here. The detected vector S' of S is then taken as the selected vector of S associated with the smallest distance $\|R' - SY\|$. However, only the first component s_1' of S' is accepted as the detected value of the data symbol s_1. The remainning n-1 components of S' are then stored and used for the next detection process. This completes the detection process for s_1 from R', and the detector continues to operate in this way for the next detection process.

The structures and operations of the multi-level threshold device shown in Fig. 4.2 is now described. Assuming the correct detection of the g previous data symbols so that the vector R' is as given by eqn. 4.2, the value of e_h in eqn. 4.4 now becomes

$$\begin{aligned} e_h &= R'Z_h^T \\ &= SYZ_h^T + WZ_h^T \end{aligned} \tag{4.5}$$

for $h = 1, 2, \cdots, f$ where f is the total number of networks $\{Z_h^T\}$ shown in Fig. 4.2, and SY and W are the received signal vector and the noise vector respectively. Furthermore, let e_h' be a possible value of e_h obtained in the absence of noise. That is, W = 0 and

$$e_h' = SYZ_h^T \tag{4.6}$$

for $h = 1, 2, \cdots, f$. Clearly, for a given value of h, there are altogether 2^n values of $\{e_h'\}$ corresponding to the 2^n possible vectors of SY or S. These 2^n values of $\{e_h'\}$ are in fact evaluated and listed in order of magnitudes in this system before the transmission of data begins. Thus, the multi-level threshold device shown in Fig. 4.2 operates by comparing

the value of e_h (eqn. 4.5) obtained at the output of the network Z_h^T (Fig. 4.2) with some threshold values, to select a total of n_v neighbouring values of $\{e_h'\}$ (eqn. 4.6). These threshold values are such that, every n_v neighbouring values of $\{e_h'\}$ are separated by a threshold value which is selected to be half way between the adjacent values of $\{e_h'\}$. Thus, if there are altogether n_t threshold values, then

$$2^n = (n_v)\ (n_t + 1) \qquad (4.7)$$

The process of selecting n_v neighbouring values of $\{e_h'\}$ from the value of e_h here is obviously the same as the analog to digital conversion process described previously where a value r (corresponding to e_h here) is required to be coded into a possible value of q (corresponding to the n_v neighbouring $\{e_h'\}$ here). Consequently, it can be seen that, only $\log_2(n_t+1)$ operations are required to select the n_v neighbouring values of $\{e_h'\}$ here, and each operation involves just a comparison of the value of e_h with a threshold value. In practice, the threshold device here selects and gives at its output the corresponding n_v possible vectors of S, instead of the n_v neighbouring values of $\{e_h'\}$ mentioned above, bearing in mind that the 2^n values of $\{e_h'\}$ here are actually calculated from the corresponding 2^n possible vectors of S (eqn. 4.6).

Thus, it can be seen from the descriptions given above that, before the transmission of data begins, n_t threshold values are required to be evaluated and stored together with the corresponding (n_t+1) groups of possible vectors $\{S\}$, for each of the f multi-level threshold devices shown in Fig. 4.2. Each of the (n_t+1) groups of possible vectors $\{S\}$ here consists of n_v possible vectors of S having the neighbouring values of $\{e_h'\}$. During the detection process, however, each threshold device here operates by comparing the value of e_h (eqn. 4.5) with the appropriate $\log_2(n_t+1)$ threshold values to select and give at its output the

appropriate n_v possible vectors $\{S\}$, which obviously can be carried out very simply and rapidly.

The value of n_v to be used in the detection process here is now discussed. It can be seen from eqns. 3.5, 3.9, and 4.6 that

$$\begin{aligned} e'_h &= SYZ_h^T \\ &= s_1(Y_1 Z_h^T) + s_2(Y_2 Z_h^T) + \cdots + s_n(Y_n Z_h^T) \end{aligned} \tag{4.8}$$

for $h = 1, 2, \cdots, f$ where Y_j is the jth row of the $n \times n$ matrix Y, and $s_1, s_2, \cdots, s_n$ are the n components of the vector S. Clearly, whenever $Y_j Z_h^T = 0$ for any possible value of j, the value of e'_h becomes independent of the value of s_j. Since there are two possible values of s_j, there are now at least two possible vectors of S having the same value of e'_h. This may, in fact, also happen even if $Y_j Z_h^T \neq 0$. Furthermore, it can be seen that if there are more than n_v possible vectors of S having the same value of e'_h, then there are at least two identical threshold values in the threshold device of Fig. 4.2, bearing in mind that every n_v neighbouring values of $\{e'_h\}$ are separated by a threshold value here. In order to ensure that every threshold has a different value, the value of n_v to be used in the detection process must now be such that

$$n_v \geq n_m \tag{4.9}$$

where n_m is the maximum number of possible vectors $\{S\}$ having the same value of e'_h, for all possible values of h.

Having described the operations of the detection process here, it will be interesting to see the implications of these operations. This is discussed as follows. It can be seen from eqns. 4.5 and 4.6 that, e_h is the orthogonal projection of the vector R' onto the unit vector Z_h, and

$\{e'_h\}$ are the orthogonal projections of the corresponding possible vectors of SY onto the same vector Z_h, in the n-demensional Euclidean vector space containning R' and SY. Similarly, the threshold values used in the threshold device shown in Fig. 4.2 may be regarded as the orthogonal projections of the appropriate hyperplanes onto Z_h in the same vector space, where these hyperplanes are perpendicular to the direction of Z_h. Thus, if there are n_t threshold values, then the n-dimensional Euclidean vector space here is effectively partitioned into (n_t+1) subspaces by the n_t hyperplanes which are perpendicular to a specific direction given by the vector Z_h. Each of these (n_t+1) subspaces contains n_v possible vectors of SY and hence n_v values of $\{e'_h\}$, bearing in mind that there are n_v values of $\{e'_h\}$ between any adjacent threshold values used in the threshold device here. An example of the descriptions given above is shown in Fig. 4.3 where $n = 4$, $n_t = 7$, and $n_v = 2$. The operation of the threshold device shown in Fig. 4.2 is therefore to determine the subspace in which the vector R' lies, and to select the n_v possible vectors of S corresponding to the n_v vectors of SY that lie in the same subspace, by comparing the value of e_h (orthogonal projection of R' onto Z_h) with some threshold values (hyperplanes perpendicular to Z_h). Consequently, the detection process here (Fig. 4.2) effectively performs a simple partitionning of the n-dimensional Euclidean vector space containning R' and SY, by using some hyperplanes that are all perpendicular to a specific direction given by the vector Z_h. A simple search process is then carried out to determine the subspace in which R' lies and to select the possible vectors of S corresponding to the vectors of SY that lie in the same subspace.

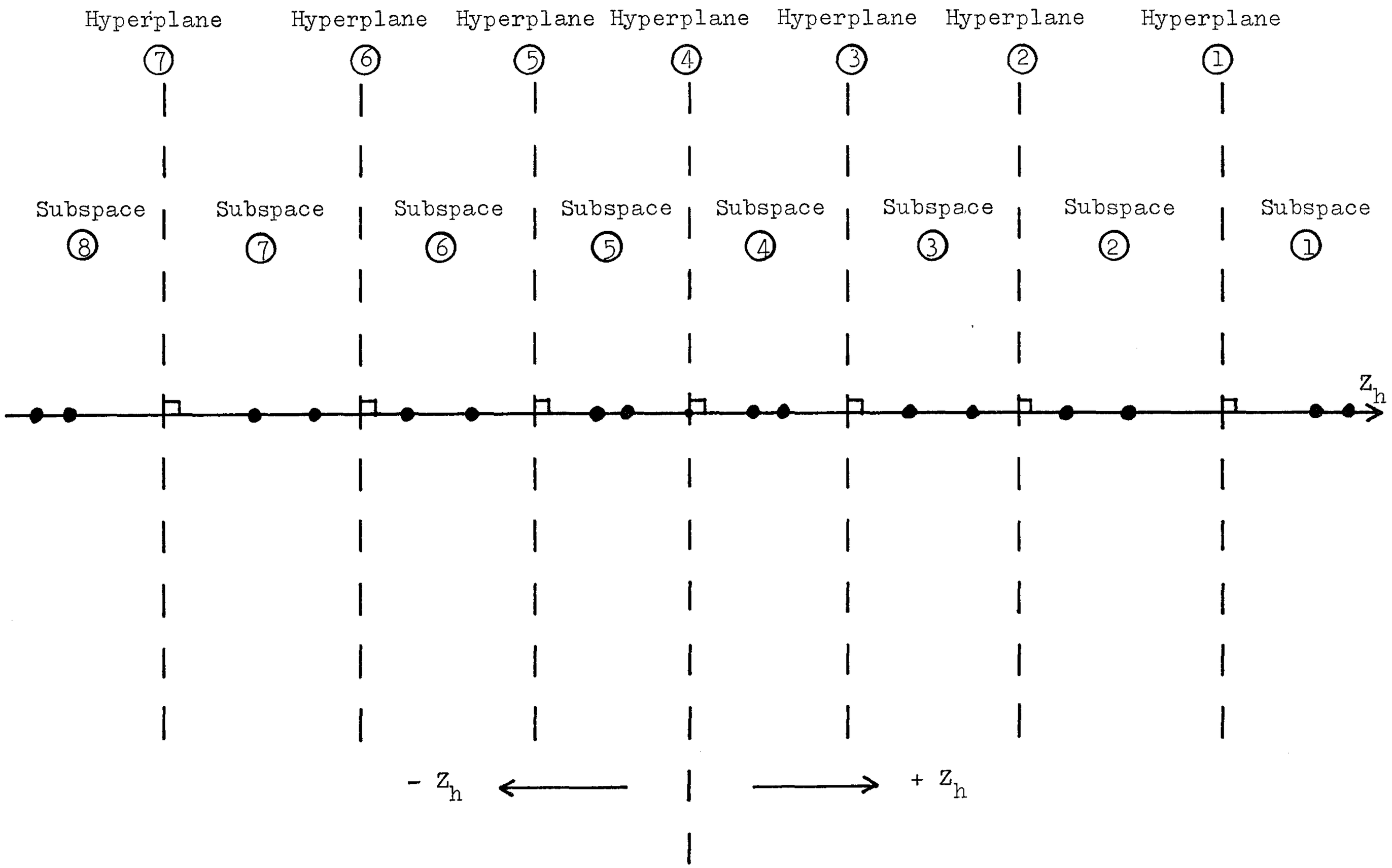


Fig. 4.3 n_t hyperplanes partitionning a n-dimensional Euclidean vector space into $n_t + 1$ subspaces. Each subspace contains n_v possible vectors of $\{SY\}$ whose orthogonal projections $\{e'_h\}$ onto the unit vector Z_h are represented by ● .
$n = 4$, $n_t = 7$, $n_v = 2$.

Having described the operations and some properties of the detection process emplying the arrangement of Fig. 4.2, the remainning part of this chapter is mainly concerned with the selections of the networks or filters $\{Z_h^T\}$ to be used here. Thus, each of the systems 1 - 6 to be developed and studied in the following sections employs the same arrangement of Fig. 4.2 in the detection of S from R', but with different $\{Z_h^T\}$. Computer simulation tests are first carried out to determine the tolerance to additive white Gaussian noise of each of these systems operating over channel A (Table 2.1). The more promising systems are then further tested on channels B and C (Table 2.1).

4.5 System 1

In system 1, the networks $\{Z_h^T\}$ in Fig. 4.2 are selected to be the filters that are matched to the vectors $\{Y_h\}$, where Y_h is the n-component vector defined by eqn. 3.9. That is,

$$Z_h = \frac{Y_h}{\|Y_h\|} \tag{4.10}$$

for $h = 1, 2, \cdots, f$ where f may have any integer value between 1 to n inclusive, and $\|Y_h\|$ is the Euclidean norm or length of the vector Y_h. The inclusion of $\|Y_h\|$ in eqn. 4.10 is to ensure that eqn. 4.3 is satisfied so that Z_h is now an unit vector.

It can be seen from eqns. 4.5 and 4.8 that, the value of e_h obtained at the output of the network Z_h^T shown in Fig. 4.2 is now given by

$$\begin{aligned} e_h &= s_1(Y_1 Z_h^T) + s_2(Y_2 Z_h^T) + \cdots + s_n(Y_n Z_h^T) + WZ_h^T \\ &= s_1\frac{(Y_1 Y_h^T)}{\|Y_h\|} + s_2\frac{(Y_2 Y_h^T)}{\|Y_h\|} + \cdots + s_n\frac{(Y_n Y_h^T)}{\|Y_h\|} + WZ_h^T \end{aligned} \tag{4.11}$$

for $h = 1, 2, \cdots, f$ where WZ_h^T is the noise component of e_h, and $s_1, s_2, \cdots, s_n$ are the n components of the vector S to be detected here. It can be shown[A9] that, under the various conditions assumed here, the noise components $\{WZ_h^T\}$ are the Gaussian random variables with zero mean and a fixed variance σ^2 where σ^2 is the two-sided power spectral density of the noise added at the output of the transmission path shown in Fig. 2.1. Furthermore, $Y_j Z_h^T$ is the orthogonal projection of the vector Y_j onto the unit vector Z_h, and it has a maximum magnitude when Z_h is as given by eqn. 4.10.[A9] It therefore follows that, for a given noise variance σ^2, the network Z_h^T used in system 1 effectively maximizes the magnitude of the component of s_h in e_h, bearing in mind that it is required to determine the detected values of $s_1, s_2, \cdots, s_n$ in the detection process here (Fig. 4.2).

The value of n_m for system 1 is now discussed, where n_m is the maximum number of possible vectors $\{S\}$ having the same value of e_h' for all possible values of h. The value of e_h' for system 1 is, from eqns. 4.6, 4.8, and 4.10,

$$e_h' = s_1 \frac{(Y_1 Y_h^T)}{\|Y_h\|} + s_2 \frac{(Y_2 Y_h^T)}{\|Y_h\|} + \cdots + s_n \frac{(Y_n Y_h^T)}{\|Y_h\|} \qquad (4.12)$$

for $h = 1, 2, \cdots, f$. Thus, for a given value of h, there are altogether 2^n values of $\{e_h'\}$ corresponding to the 2^n possible vectors $\{S\}$ having the 2^n possible combinations of the n components $s_1, s_2, \cdots, s_n$. However, it can be seen from eqn. 3.9 that,

$$Y_j Y_h^T = 0 \qquad \text{whenever} \qquad |j - h| > g \qquad (4.13)$$

for any positive integer values of j and h, where g+1 is the number

of components of the channel sampled impulse response, and $|j - h|$ is the absolute value of $j - h$. This implies that, some of the components in e'_h for system 1 (eqn. 4.12) are now equal to zero whenever $n > g+1$. For example, if $n = 8$ and $g = 4$, then it can be seen from eqns. 4.12 and 4.13 that the components of s_6, s_7, and s_8 in e'_1 (eqn. 4.12) are now equal to zero. Consequently, there are now 2^3 possible vectors $\{S\}$ (those with the 2^3 different combinations of s_6, s_7, and s_8) having the same value of e'_1. It is, of course, possible for some of the possible vectors $\{S\}$ to have the same value of e'_h even if $Y_j Y_h^T \neq 0$. The value of n_m in the example given above is therefore at least as large as 2^3. Table 4.1 shows the values of $\{n_m\}$ for system 1 operating over channel A (Table 2.1). Each of these values is obtained by inspecting all the 2^n values of $\{e'_h\}$ for $h = 1, 2, \cdots, f$, where f is the number of the networks $\{Z_h^T\}$ shown in Fig. 4.2. The values of $\{e'_h\}$ here are, of course, calculated by using eqn. 4.12.

	f = 1	f = 2	f = 3	f = 4	f = 5	f = 6	f = 7	f = 8
n = 8	8	8	8	8	8	8	8	32
n = 5	1	2	2	2	4			
n = 3	1	1	1					

Table 4.1 Maximum number (n_m) of possible vectors $\{S\}$ having the same value of e'_h ($1 \leq h \leq f$) (eqn. 4.12) for system 1 operating over channel A (Table 2.1).

The tolerance to noise of system 1 operating over channel A (Table 2.1) has been studied by using computer simulation tests. Various values of n, f, and n_v have been used in the tests, where f is the number of the networks $\{Z_h^T\}$ shown in Fig. 4.2, and n_v is the number of possible vectors $\{S\}$ selected at the output of each threshold device shown in Fig. 4.2. For given values of n and f, the value of n_v used here is also selected to be at least as large as the corresponding value of n_m shown in Table 4.1, where n_m is, of course, the maximum number of possible vectors $\{S\}$ having the same value of e'_h given by eqn. 4.12. This is to ensure that all the threshold values in each threshold device of Fig. 4.2 are different, as is explained in section 4.4. The results of the computer simulation tests are shown in Figs. 4.4 - 4.6 where the 95% confidence limits of the curves are about ± 0.5 dB. The bit error rate P_e and the signal to noise ratio SNR here are as defined by eqns. 3.39 and 3.40 respectively. The optimum system is the system described in section 4.3.

Fig. 4.4 shows the performance of system 1 with $n = 8$. As it appears, the tolerance to noise of the system is significantly improved by increasing the value of f (number of networks $\{Z_h^T\}$ used in Fig. 4.2). When $f = 5$ and $n_v = 8$, the tolerance to noise of this system is about 1 dB lower than that of the optimum system at an error rate of 10^{-3}. The number of distance measurements required here, being equal to (fn_v+2), is 42 and is a small number as compared to that (2^n or 256) required in the optimum system. The distance associated with a vector S, is of course the quantity $\|R' - SY\|$.

Fig. 4.5 shows the performance of system 1 with $n = 5$. It can be seen that, the tolerance to noise of the optimum system with $n = 5$ is about 1 dB lower than that of the optimum system with $n = 8$ at an

error rate of 10^{-3}, and the difference approaches to be insignificant at error rates below 10^{-4}. Consider the curve for system 1 with f = 1 and n_v = 2. It appears that the improvement in the tolerance to noise gained by doubling the value of n_v is larger than that gained by tripling the value of f, bearing in mind that the total number of the possible vectors $\{S\}$ selected for distance measurements is here given by (fn_v+2). One possible reason to this is that, while the n_v possible vectors $\{S\}$ selected in one threshold device of Fig. 4.2 are all different, some of the possible vectors $\{S\}$ selected in more than one threshold device (that is f > 1) may be corresponding to the same vector. Consequently, at the outputs of all the f threshold devices shown in Fig. 4.2, there are now 4 different possible vectors $\{S\}$ in the system with f = 1 and n_v = 4, whereas there are probably less than 4 different possible vectors $\{S\}$ in the system with f = 3 and n_v = 2. It can also be seen from Fig. 4.5 that the optimum performance for n = 5 can be achieved by system 1 with f = 3 and n_v = 4 which involves not more than fn_v+2 or 14 distance measurements. The performance of this system with f = 1 and n_v = 8 also appears to approach that of the optimum system. The number of distance measurements required here is 10, as compared to that (2^n or 32) required in the optimum system.

The performance of system 1 with n = 3 is shown in Fig. 4.6. The optimum system here involves 2^n or 8 distance measurements, and it has a tolerance to noise of about 2.5 dB more inferior in relation to that of the optimum system with n = 8 at an error rate of 10^{-3}. As it appears, the optimum performance here (that is, for n = 3) can be achieved by system 1 with f = 1 and n_v = 2 which involves just fn_v+2 or 4 distance measurements.

It can be seen from the results shown in Figs. 4.4 - 4.6 that, for given values of f and n_v and at high signal to noise ratios where the error rates drop below 10^{-3}, system 1 appears to perform better when a smaller value of n is used. Thus, by increasing the value of n, the total number of possible vectors $\{S\}$ (being given by 2^n) also increases and more vectors of $\{S\}$ are now required to be considered (for distance measurements) in the detection process of system 1 to achieve a given tolerance to noise. However, the weakness of using a smaller value of n is that the optimum performance achievable by the system is also reduced. Consequently, a moderate value of n should be used in system 1, so that a high tolerance to noise may be achieved here with a managable number of distance measurements involved in the system. As it appears, n = 5 is the most promising of all the three values of n tested in Figs. 4.4 - 4.6, for system 1 operating over channel A (Table 2.1). In particular, system 1 with n = 5, f = 1, and n_v = 8 appears to be very promising as it gives a near optimum performance while requiring just 10 distance measurements in its detection process.

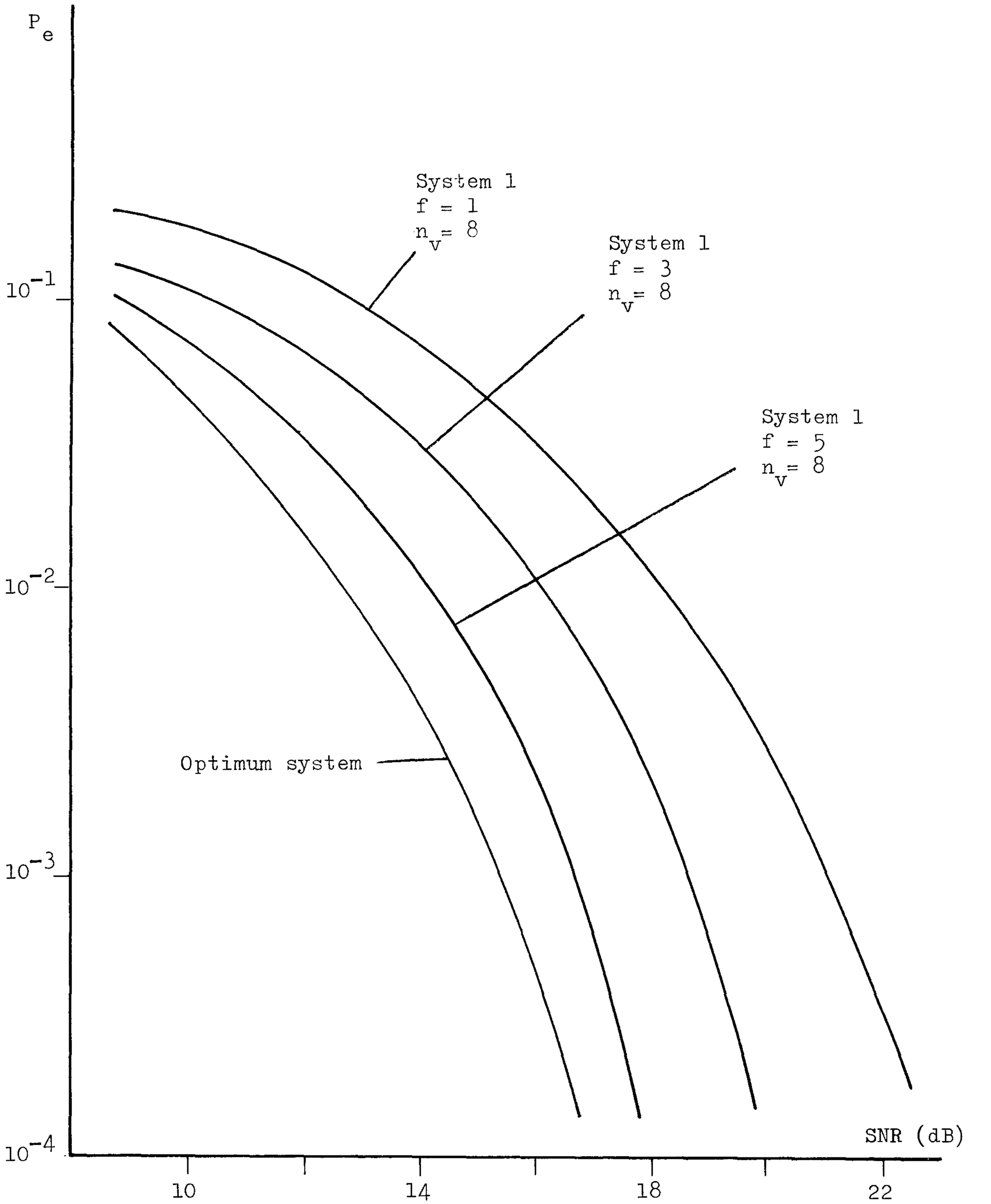


Fig. 4.4 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 1 operating over channel A (Table 2.1).

Number of samples used in the system (n) = 8.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

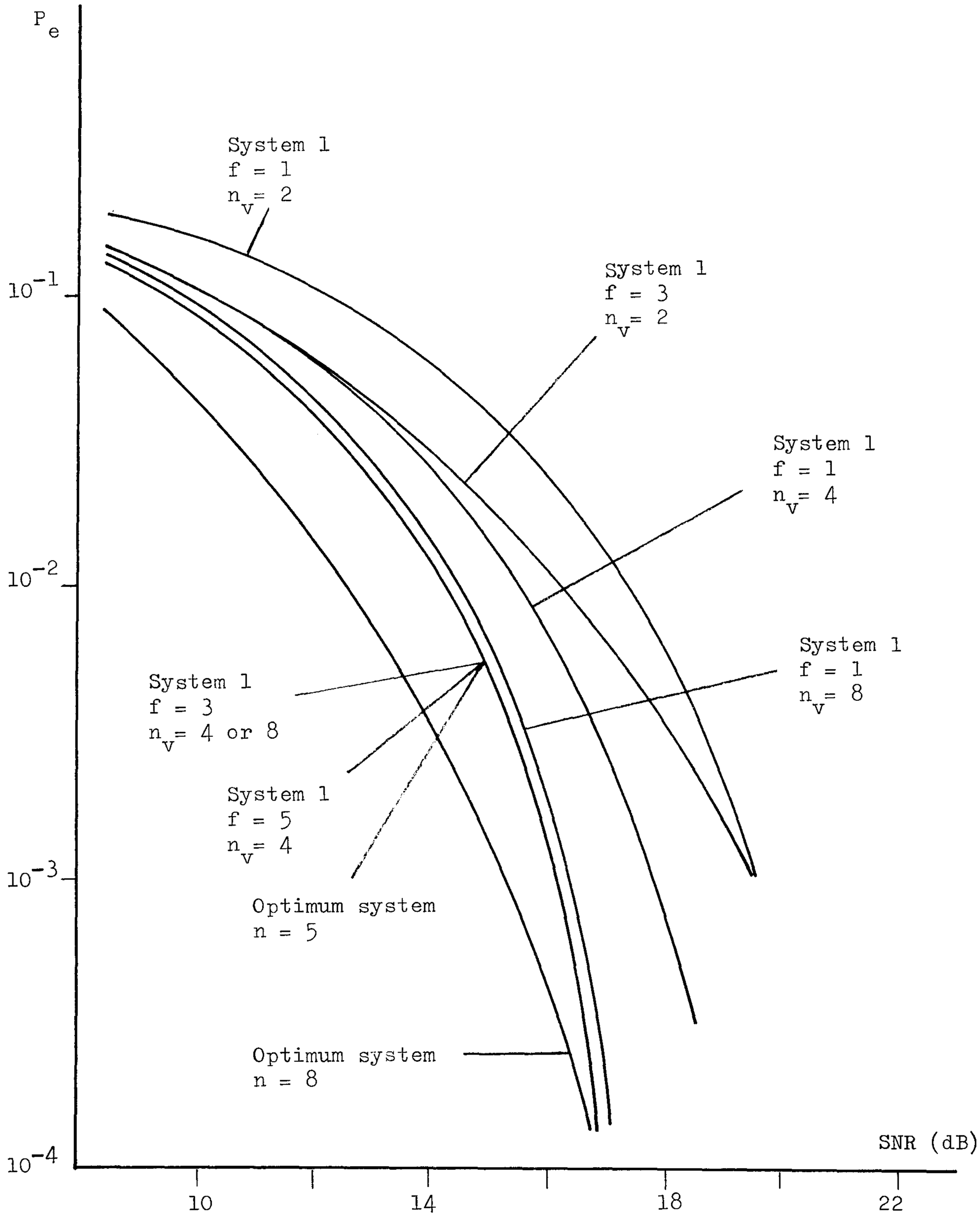


Fig. 4.5 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 1 operating over channel A (Table 2.1).

Number of samples used in the system (n) = 5.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

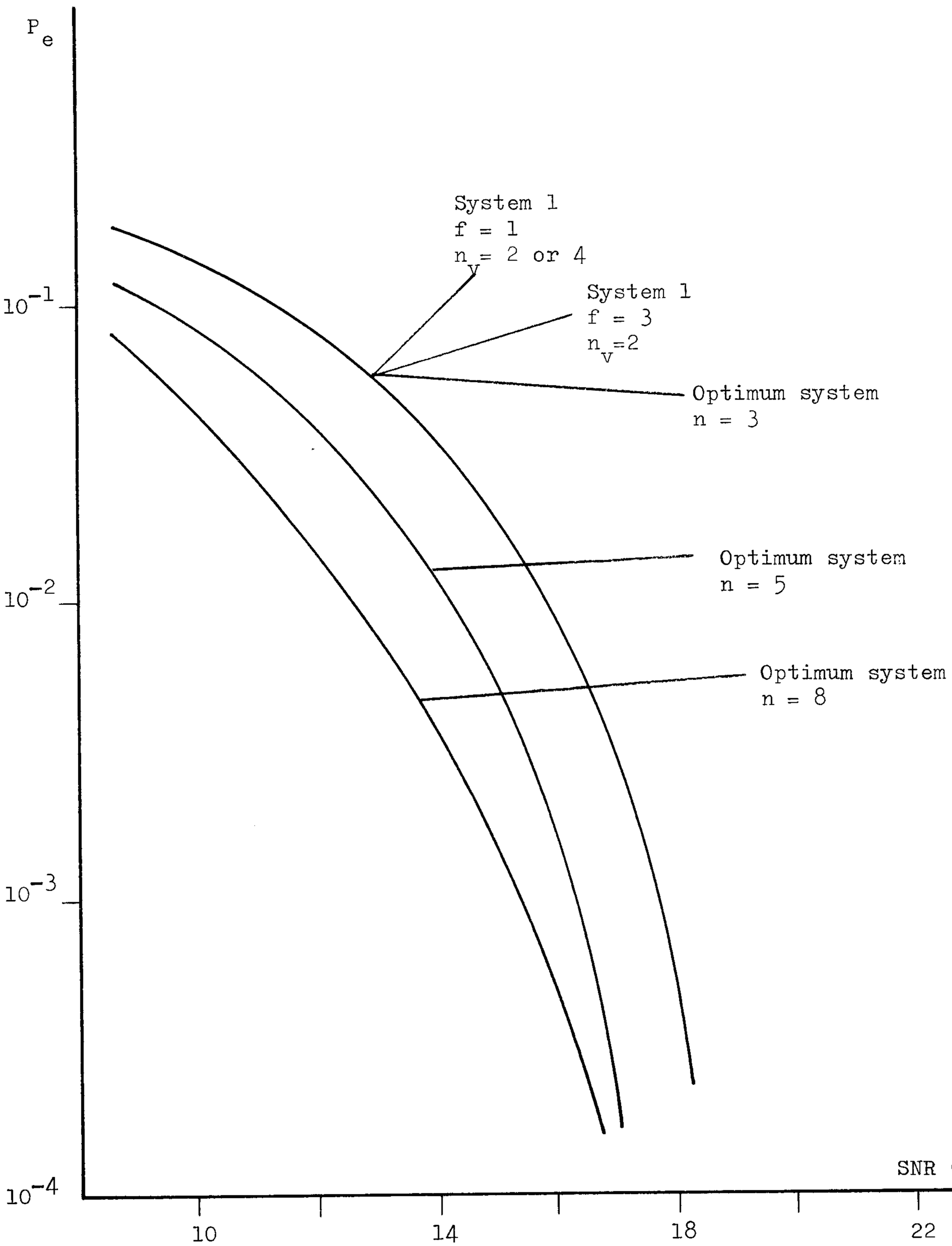


Fig. 4.6 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 1 operating over channel A (Table 2.1).

Number of samples used in the system (n) = 3.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

4.6 System 2

In system 2, the f filters $\{Z_h^T\}$ shown in Fig. 4.2 are selected to be given by a set of f orthonormal vectors $\{Z_h\}$. That is,

$$\begin{aligned} Z_j Z_k^T &= 0 \qquad \text{if} \quad j \neq k \\ &= 1 \qquad \text{if} \quad j = k \end{aligned} \tag{4.14}$$

for any possible values of j and k. The Gram-Schmidt orthogonalisation process used in section 3.12 is used here to derive these n-component vectors. Thus, from eqn. 3.65, a set of f n-component vectors is first evaluated as

$$\begin{aligned} Z'_1 &= Y_1 \\ Z'_2 &= Y_2 - \frac{1}{\|Z'_1\|^2} Y_2 (Z'_1)^T Z'_1 \\ &\vdots \\ Z'_f &= Y_f - \frac{1}{\|Z'_{f-1}\|^2} Y_f (Z'_{f-1})^T Z'_{f-1} - \cdots - \frac{1}{\|Z'_1\|^2} Y_f (Z'_1)^T Z'_1 \end{aligned} \tag{4.15}$$

where $\|Z'_h\|$ is the Euclidean norm or length of the vector Z'_h, and $(Z'_h)^T$ is the transpose of the row matrix Z'_h, bearing in mind that a vector is also treated as a row matrix in this thesis. The $\{Y_j\}$ here are the n-component vectors defined by eqn. 3.9. The f n-component vectors $\{Z_h\}$ used in system 2 are then evaluated as

$$Z_h = \frac{Z'_h}{\|Z'_h\|} \tag{4.16}$$

for $h = 1, 2, \cdots, f$.

This set of orthonormal vectors $\{Z_h\}$ has the interesting property given by eqn. 3.70, and that is

$$Y_j Z_h^T = 0 \qquad \text{if} \qquad j < h \tag{4.17}$$

for any possible values of j and h. The value of e_h' (eqns. 4.6 and 4.8) in system 2 is therefore given by

$$e_h' = s_h(Y_h Z_h^T) + s_{h+1}(Y_{h+1} Z_h^T) + \cdots + s_n(Y_n Z_h^T) \tag{4.18}$$

for $h = 1, 2, \cdots, f$. Clearly, the value of e_h' here is independent of the values of the first h-1 components $s_1, s_2, \cdots, s_{h-1}$ of the data-symbol vector S. It follows that there are now 2^{h-1} possible vectors $\{S\}$ (with the 2^{h-1} possible combinations of $s_1, s_2, \cdots, s_{h-1}$) having the same value of e_h'. The maximum number of the possible vectors $\{S\}$ having the same value of e_h' for any possible value of h is now at least as large as 2^{f-1}, bearing in mind that f is the largest possible value of h here. Furthermore, it is recalled that the number of possible vectors $\{S\}$ selected at the output of each threshold device shown in Fig. 4.2 is n_v and is constrained to be at least as large as the value of n_m (eqn. 4.9), where n_m is the maximum number of possible vectors $\{S\}$ having the same value of e_h' for any possible value of h. It therefore follows that the value of n_v to be used in system 2 must be at least as large as 2^{f-1}. Consequently, the value of f (number of networks $\{Z_h^T\}$) used in system 2 must be kept to a small value so that the number of distance measurements, being given by (fn_v+2), involved in the system is kept to a managable value. The values of $\{n_m\}$ for system 2 operating over channel A (Table 2.1) are shown in Table 4.2. Each of these values is obtained by inspecting all the 2^n calculated values of $\{e_h'\}$ corresponding

to the 2^n possible vectors $\{S\}$, for $h = 1, 2, \cdots f$. The calculations here are carried out by using eqns. 4.8, 4.15, and 4.16.

Computer simulation tests have been carried out to determine the tolerance to noise of system 2 operating over channel A (Table 2.1), and the results are shown in Figs. 4.7 - 4.9. The 95% confidence limits of these results are about ± 0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and 3.40 respectively. The optimum system here is, of course, the system described in section 4.3.

Comparing the results shown in Figs. 4.4 - 4.9, it is observed that, given the values of n, f, and n_v, system 2 appears to have about the same performance over channel A as that of system 1. In particular, whenever $f = 1$, exactly the same performance is achieved by both systems 1 and 2. This is because both systems here use the same vector for Z_1 as can be seen from eqns. 4.10, 4.15, and 4.16. The most promising arrangement of system 2 is the same as that of system 1 and is when $n = 5$, $f = 1$, and $n_v = 8$.

	f = 1	f = 2	f = 3	f = 4	f = 5	f = 6	f = 7	f = 8
n = 8	8	8	8	8	16	32	64	128
n = 5	1	2	4	8	16			
n = 3	1	2	4					

Table 4.2 Maximum number (n_m) of possible vectors $\{S\}$ having the same value of e'_h ($1 \leq h \leq f$) (eqn. 4.18) for system 2 operating over channel A (Table 2.1).

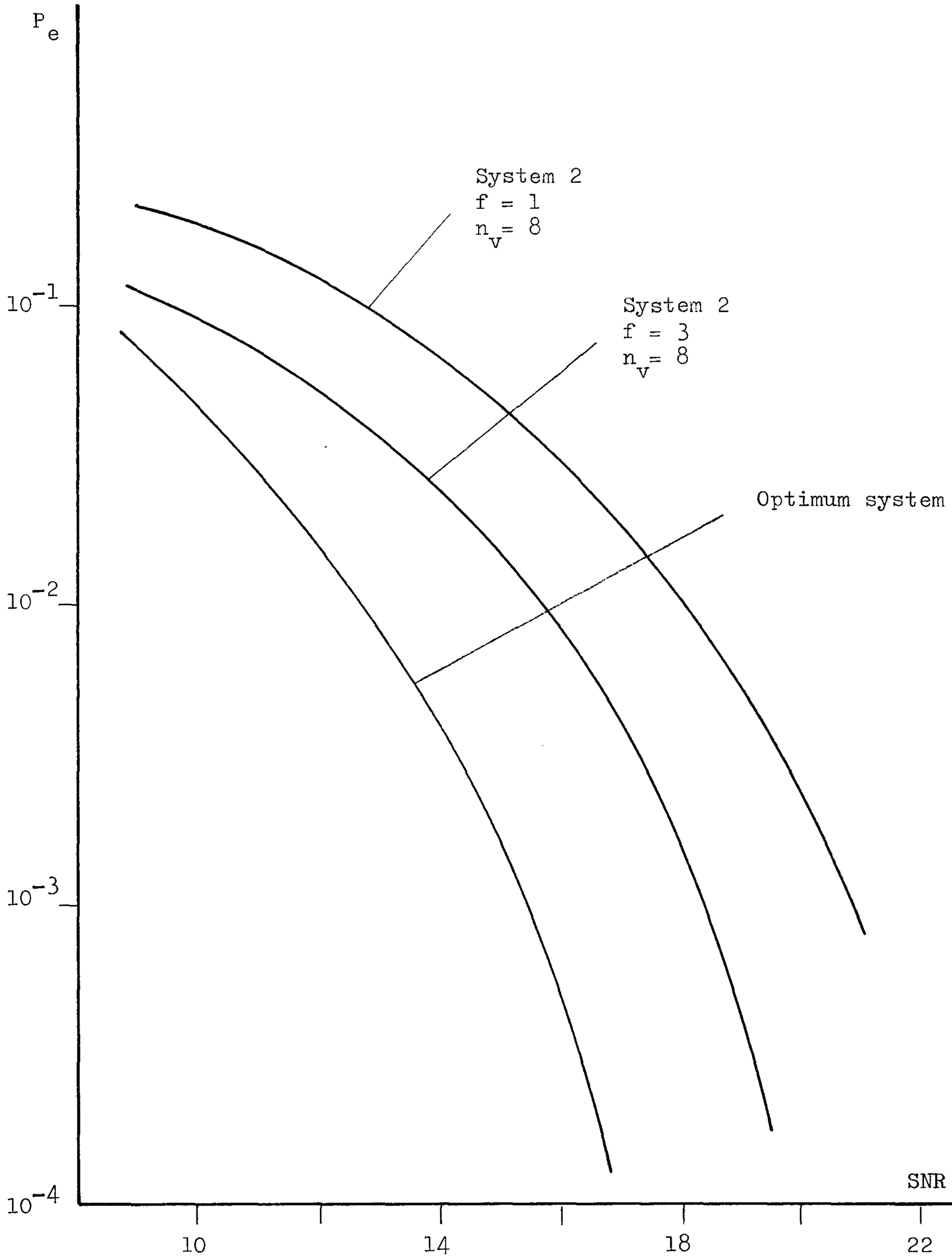


Fig. 4.7 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 2 operating over channel A (Table 2.1).

Number of samples used in the system (n) = 8.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

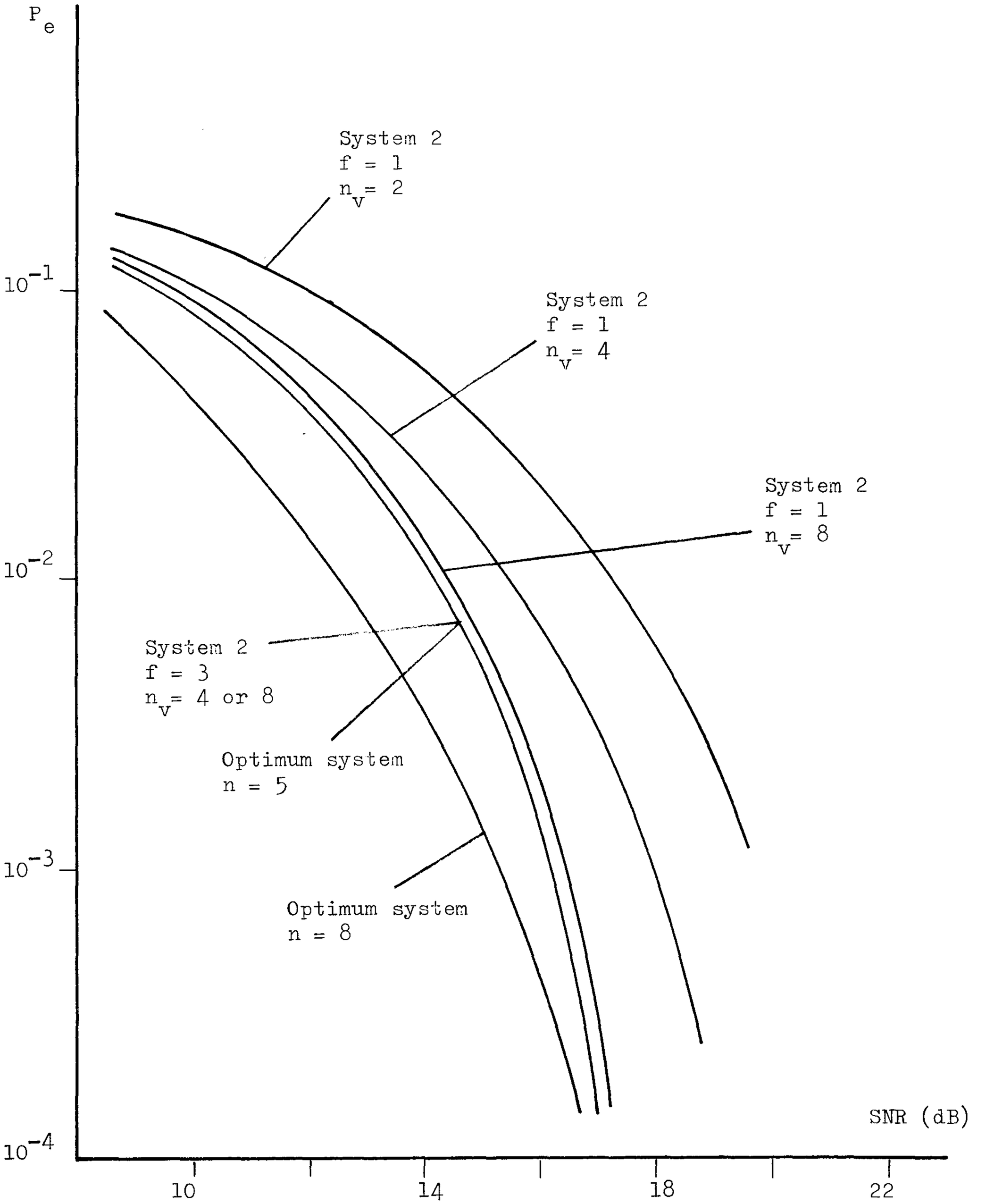


Fig. 4.8 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 2 operating over channel A (Table 2.1).

Number of samples used in the system (n) = 5.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

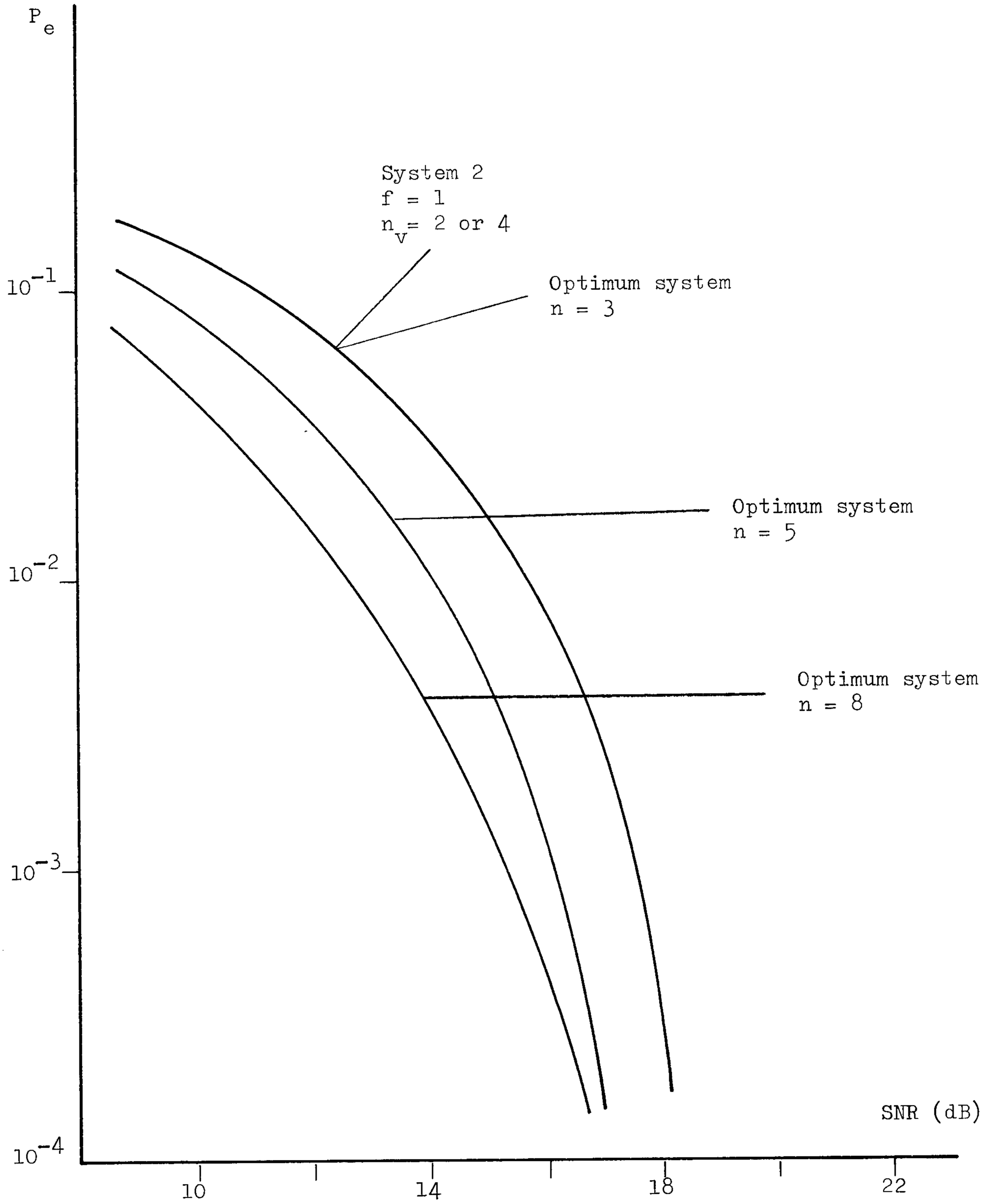


Fig. 4.9 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 2 operating over channel A (Table 2.1).

Number of samples used in the system (n) = 3.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

So far, both systems 1 and 2 have involved the use of one or more filters $\{Z_h^T\}$ (that is, $f \geq 1$) in the detection process for the vector S from R' using the arrangement shown in Fig. 4.2. Each filter here, say Z_h^T, is implemented as a n-tap transversal filter and so n multiplications are involved here in the evaluation of the value of e_h at its output. Furthermore, some of the possible vectors $\{S\}$ obtained at the output of one threshold device (Fig. 4.2) may be exactly the same as some of those obtained at the output of another threshold device here. It therefore follows that, in order to reduce the number of operations and to make use of every possible vector of S obtained at the output of the threshold device, just one filter Z_1^T should be used in the arrangement of Fig. 4.2. This is the arrangement used in each of the following systems.

4.7 System 3

In system 3, only one filter Z_1^T is used in the detection process for S from R' using the arrangement of Fig. 4.2. This filter is derived as follows. Let D be the n-component row matrix defined as

$$D = Z_1 Y^T \tag{4.19}$$

where Z_1 is the n-component row matrix (or vector) for the filter Z_1^T shown in Fig. 4.2, and Y^T is the transpose of the $n \times n$ matrix Y defined by eqn. 3.9. The value of e_1' in eqn. 4.6 now becomes

$$e_1' = SD^T \tag{4.20}$$

where D^T is the transpose of D, and S is the n-component row matrix (or vector) to be determined in the detection process here (Fig. 4.2).

It should perhaps be reminded that, the value of e_1' is a possible value of e_1 obtained, in the absence of noise, at the output of the filter Z_1^T shown in Fig. 4.2. Furthermore, the detection process here (Fig. 4.2) selects, for distance measurements, the n_v possible vectors of S corresponding to the n_v neighbouring values of e_1' (eqn. 4.20) which lie between the same adjacent threshold values as those of the value of e_1 (Fig. 4.3). In the selection of the filter Z_1^T for system 3, the n-component row matrix D in eqn 4.20 is first selected to be such that, all the 2^n values of $\{e_1'\}$ corresponding to the 2^n possible vectors $\{S\}$ are different and are uniformly spaced. That is,

$$D = c \left[2^{n-1} \quad 2^{n-2} \quad \cdots \quad 2^1 \quad 2^0 \right] \tag{4.21}$$

where c is a constant to be discussed shortly. The value of e_1' in eqn. 4.20 now becomes

$$e_1' = c \left(2^{n-1}s_1 + 2^{n-2}s_2 + \cdots + 2s_{n-1} + s_n \right) \tag{4.22}$$

where the n components $s_1, s_2, \cdots, s_n$ of S can have any of the two binary values 1 and -1. It can be seen from eqn. 4.22 that, every adjacent values of $\{e_1'\}$ are now separated by the same amount of 2c. Having set the row matrix D to that given by eqn. 4.21, the filter Z_1^T to be used in system 3 can now be determined, from eqn. 4.19, as

$$Z_1 = D\,(Y^T)^{-1} \tag{4.23}$$

where $(Y^T)^{-1}$ is the inverse of the $n \times n$ matrix Y^T. The value of c in eqn. 4.21 is here selected to be such that $Z_1 Z_1^T = 1$, so that Z_1 is now an unit vector. Thus, the selection of the value of c and the evaluation of Z_1 with eqn. 4.23 can be carried out as follow. A row

matrix is first evaluated as

$$Z_1' = c^{-1} (D) (Y^T)^{-1}$$
$$= \left[2^{n-1} \quad 2^{n-2} \quad \cdot\cdot \quad 2^0 \right] (Y^T)^{-1} \qquad (4.24)$$

The unit vector Z_1 is then evaluated as

$$Z_1 = \frac{Z_1'}{\| Z_1' \|} \qquad (4.25)$$

where $\| Z_1' \|$ is the Euclidean norm or length of the vector Z_1', bearing in mind that a row matrix is also treated as a vector in this thesis. Consequently, in selecting the filter Z_1^T to be used in system 3, it involves just the use of eqns. 4.24 and 4.25. It should be noticed that it is not necessary to evaluate the value of c in the row matrix D given by eqn. 4.21 here.

Computer simulation tests have been carried out to determine the tolerance to noise of system 3 operating over channel A whose sampled impulse response is as given in Table 2.1. The results of these tests are shown in Fig. 4.10 where the 95% confidence limit of each curve is about ± 0.5 dB. The bit error rate P_e and the signal to noise ratio SNR here are as defined by eqns. 3.39 and 3.40 respectively. Fig. 4.10 also shows the performance of the optimum system described in section 4.3.

It can be seen from Fig. 4.10 that, for a given value of n_v (number of possible vectors $\{S\}$ selected at the output of the threshold device shown in Fig. 4.2), system 3 with n = 8 appears to have the poorest performance as compared to that of the given system with n = 5 or n = 3. The best performance of system 3 here is achieved when n = 5 and $n_v = 4$. However, this performance is far worse than the best performance achieved

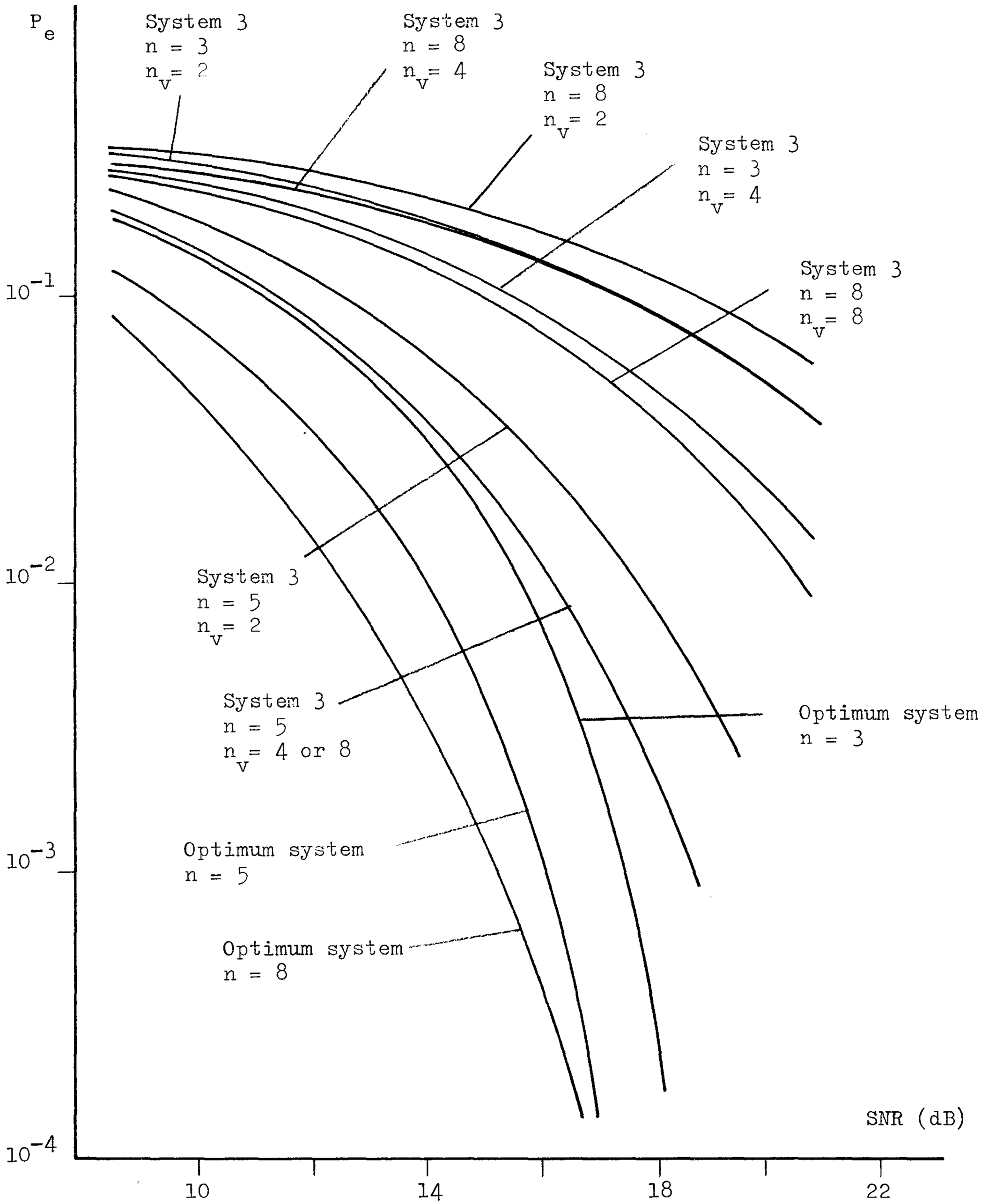


Fig. 4.10 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 3 operating over channel A (Table 2.1).

Number of of filters used in the system (f) = 1 (Fig. 4.2).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

by any of the systems 1 and 2 with f = 1, as can be seen from Figs. 4.4 - 4.10. The difference here is about 2.5 dB at an error rate of 10^{-3}, and it increases as the error rate decreases further.

Clearly, one weakness of system 3 is that, no prior knowledge of the channel sampled impulse response is used in setting the row matrix D (eqn. 4.21). In each of the following systems, the filter Z_1^T shown in Fig. 4.2 is selected to avoid this weakness of system 3.

4.8 System 4

In system 4, the filter Z_1^T (Fig. 4.2) is selected to be the unit vector Z_1 that makes the same acute angle with each of the n vectors $Y_1, Y_2, \cdots, Y_n$ defined by eqn. 3.9. Thus, if θ is the acute angle just mentioned, then

$$\begin{aligned} Z_1 Y_j^T &= \| Z_1 \| \| Y_j \| \cos \theta \\ &= \| Y_j \| \cos \theta \\ &= c \| Y_j \| \end{aligned} \tag{4.26}$$

for $j = 1, 2, \cdots, n$ where $c = \cos \theta$ and $\| Y_j \|$ is the Euclidean norm or length of the vector Y_j. It is reminded that a vector is also treated as a row matrix in this thesis so that Y_j may be regarded as a row matrix here and Y_j^T is now the transpose of Y_j. It can be seen from eqn. 3.9 that $Z_1 Y_j^T$ is the jth component of the row matrix D defined in eqn. 4.19 and so for system 4,

$$D = c \left[\| Y_1 \| \quad \| Y_2 \| \quad \cdots \quad \| Y_n \| \right] \tag{4.27}$$

The filter Z_1^T (Fig. 4.2) to be used in system 4 can now be selected by using eqns. 4.19 and 4.27, and this is carried out by first evaluating a row matrix Z_1' as

$$\begin{aligned} Z_1' &= c^{-1}(D)\,(Y^T)^{-1} \\ &= \left[\, \|Y_1\| \;\; \|Y_2\| \;\; \cdots \;\; \|Y_n\| \,\right] (Y^T)^{-1} \end{aligned} \tag{4.28}$$

and then setting the vector Z_1 (the filter) to $\dfrac{Z_1'}{\|Z_1'\|}$. It is, of course, not necessary to evaluate the value of c here.

It can be seen from eqns. 4.20 and 4.27 that, the value of e_1 obtained, in the absence of noise, at the output of the filter Z_1^T shown in Fig. 4.2 is now given by

$$e_1' = c\,(\, s_1\|Y_1\| + s_2\|Y_2\| + \cdots + s_n\|Y_n\| \,) \tag{4.29}$$

where the value of c here is as defined in eqn. 4.26, and $s_1, s_2, \cdots, s_n$ are the n components of the vector S. Furthermore, it can be seen from eqn. 3.9 that, the vectors $\{Y_h\}$ here are such that whenever $n > g+1$, then

$$\|Y_j\| = \|Y_k\| = \|V\| \qquad \text{for } j, k < n - g \tag{4.30}$$

where V is the $(g+1)$-component vector whose components are the channel sampled impulse response $y_0, y_1, \cdots, y_g$. It therefore follows that whenever $n > g+1$, then some of the possible vectors $\{S\}$ will give the same value of e_1', bearing in mind that $e_1' = SD^T$. Thus, for example, if $g = 4$ and $n = 6$, then from eqn. 4.30, $\|Y_1\| = \|Y_2\|$. In this case, the possible vector S with $s_1 = 1$, $s_2 = -1$ can be seen (eqn. 4.29) to give the same value of e_1' as that of the vector with $s_1 = -1$, $s_2 = 1$, all other components $s_3, s_4, \cdots, s_n$ being the same for both vectors here.

Consequently, the value of n_m (being the maximum number of possible vectors $\{S\}$ that give the same value of e_1') in system 4 must now be determined so that the value of n_v (being the number of possible vectors $\{S\}$ selected at the output of the threshold device of Fig. 4.2) here can be selected to satisfy the condition of (4.9) where $n_v \geq n_m$. Table 4.3 shows the values of n_m at three different values of n for system 4 operating over channel A (Table 2.1). Each of these values is obtained by evaluating all the 2^n values of $\{e_1'\}$ using eqn. 4.29 and comparing their values. The value of c in eqn. 4.29 can be evaluated by using eqn. 4.26, bearing in mind that Z_1 has already been determined. As Table 4.3 shows, the value of n_m is 6 when n = 8, and so the value of n_v to be used in the system with n = 8 must now be at least as large as 6.

Computer simulation tests have been carried out to determine the tolerance to noise of system 4 operating over channel A (Table 2.1), and the results are shown in Fig. 4.11. The 95% confidence limits of these results are about $\pm$0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and 3.40 respectively. The optimum system here is the system described in section 4.3.

n	n_m
8	6
5	1
3	1

Table 4.3 Maximum number (n_m) of possible vectors $\{S\}$ having the same value of e_h' (eqn. 4.29) for system 4 operating over channel A (Table 2.1).

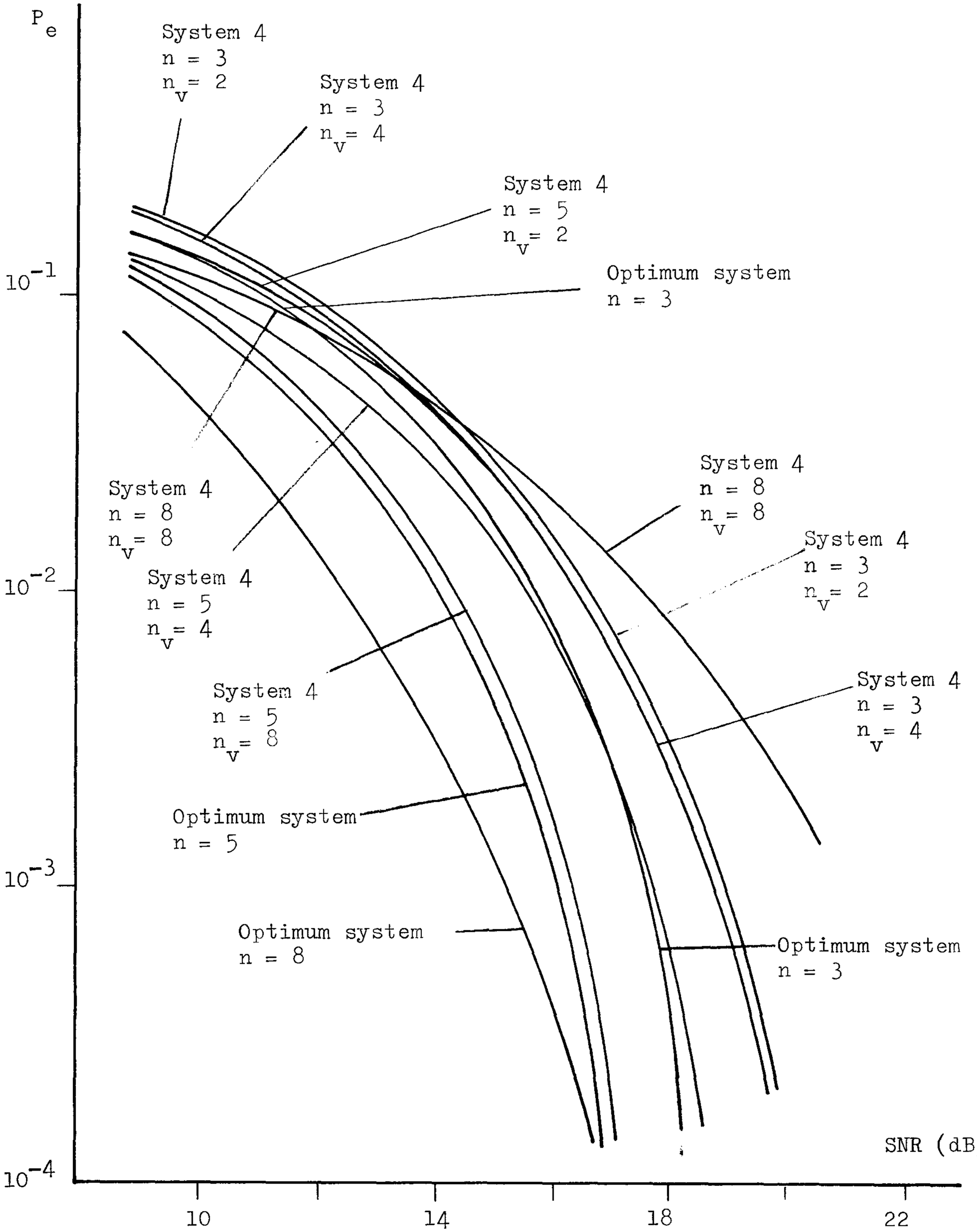


Fig. 4.11 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 4 operating over channel A (Table 2.1).

Number of filters used in the system (f) = 1 (Fig. 4.2).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

It can be seen from Fig. 4.11 that system 4 with n = 8 and $n_v = 8$ has a poor performance. The number of possible vectors $\{S\}$ considered for the distance measurements here is equal to n_v+2 or 10, and is a very small value as compared to the total number (2^8 or 256) of possible vectors $\{S\}$ here. When n = 3, the tolerance to noise of system 4 with $n_v = 2$ or 4 is about 1 dB inferior to that of the optimum system at an error rate of 10^{-3}. When n = 5, the tolerance to noise of system 4 with $n_v = 8$ approaches that of the optimum system, but as n_v is reduced to 4, system 4 suffers a loss of about 1.5 dB at an error rate of 10^{-3}.

System 4 therefore appears to be very promising so long as a moderate value of n and a sufficiently large value of n_v are used in the system.

4.9 System 5

In the detection of S from R' in system 1, each filter, say Z_h^T, shown in Fig. 4.2 is selected to maximize the magnitude of the component $s_h Y_h Z_h^T$ in e_h obtained at the output of the given filter, for a given noise level in e_h. In system 5, only one filter Z_1^T is used and it is selected to be such that, the sum of the squares of all the components $s_1 Y_1 Z_1^T$, $s_2 Y_2 Z_1^T$, $\cdots$, $s_n Y_n Z_1^T$ in e_1 is maximized, for a given noise level in e_1, bearing in mind that $\left[Y_n Z_1^T \;\; Y_{n-1} Z_1^T \;\; \cdots \;\; Y_1 Z_1^T \right]$ is effectively the resultant sampled impulse response of the baseband channel and filter Z_1^T, and that the $\{s_i\}$ here can only have any of the two values 1 and -1. This filter of system 5 therefore effectively maximizes the energy of the resultant sampled impulse response of the channel and filter, for a given noise level at its output. It can be shown[A9] that, the noise variance remains unchanged after the linear filtering of the vector R' through the filter Z_1^T (Fig. 4.2), so long as $Z_1 Z_1^T = 1$.

Thus, the filter Z_1^T used in system 5 is selected to maximize the quantity

$$d = \sum_{j=1}^{n} (Y_j Z_1^T)^2 \qquad (4.31)$$

with the constraint that $Z_1 Z_1^T = 1$, where $\{Y_j\}$ are, of course, the n-component row matrix defined by eqn. 3.9, and Z_1^T is the transpose of the n-component row matrix Z_1 which is to be determined here. Eqn. 4.31 can be further reduced to

$$d = Z_1 \left(\sum_{j=1}^{n} Y_j^T Y_j \right) Z_1^T$$

$$= Z_1 (A) Z_1^T \qquad (4.32)$$

where $$A = \sum_{j=1}^{n} (Y_j^T Y_j) = Y^T Y \qquad (4.33)$$

and Y^T is the transpose of the $n \times n$ matrix Y whose jth row is given by the n-component row matrix Y_j. In eqn. 4.33, A is obviously a $n \times n$ symmetric matrix. The Lagrange multipliers[E5] used in section 3.13 is now used to maximize the quantity d of eqn. 4.32 with the constraint that $Z_1 Z_1^T = 1$. This is carried out by defining a function f' to be

$$f' = d + \lambda (1 - Z_1 Z_1^T) \qquad (4.34)$$

where λ is a constant value, and d is the quantity defined in eqn. 4.32. Differentiating f' with respect to Z_1 and setting the resulting expressio to zero yields

$$Z_1 A = \lambda Z_1 \qquad (4.35)$$

where λ and Z_1 here are obviously the eigenvalue and eigenvector

of the matrix A. Substituting eqn. 4.35 back into eqn. 4.32 yields $d = \lambda$, and so the filter Z_1^T to be used in system 5 is here selected to be the eigenvector of A associated with the largest eigenvalue.

Using the vector Z_1 derived above, the values of $\{e_1'\}$ for system 5 operating over channel A (Table 2.1) have been evaluated (eqn. 4.8) and compared, where $\{e_1'\}$ are the possible values of e_1 (Fig. 4.2) obtained in the absence of noise. It is found here that each of the 2^n possible vectors $\{S\}$ has a different value of e_1', and so the value of n_m for the system here is equal to 1, bearing in mind that n_m is the maximum number of possible vectors $\{S\}$ having the same value of e_1'. Consequently, there is now effectively no constraint on the value of n_v to be used in the system, bearing in mind that n_v is the number of possible vectors $\{S\}$ selected at the output of the threshold device shown in Fig. 4.2 and it is constrained to be such that $n_v \geq n_m$ (section 4.4).

The tolerance to noise of system 5 operating over channel A (Table 2.1) has been studied by using computer simulation tests. The results of the tests are shown in Fig. 4.12 where the 95% confidence limit of each curve is about ± 0.5 dB. The bit error rate P_e and the signal to noise SNR here are as defined by eqns. 3.39 and 3.40 respectively. The optimum system here is the system described in section 4.3.

Fig. 4.12 shows that when $n = 8$, system 5 with the values of n_v tested here have very poor performances. The values of n_v tested here are 4 and 8 which are very small in relation to the total number (256) of possible vectors $\{S\}$, bearing in mind that n_v is the number of possible vectors $\{S\}$ selected (for distance measurements) at the output of the threshold device shown in Fig. 4.2. When $n = 3$, the optimum performance is achieved by system 5 with $n_v = 2$. When $n = 5$, the performance of system 5 with $n_v = 4$ appears to approach that of the optimum system. Overall, system 5 with $n = 5$ appears to be very promising when operating over channel A here.

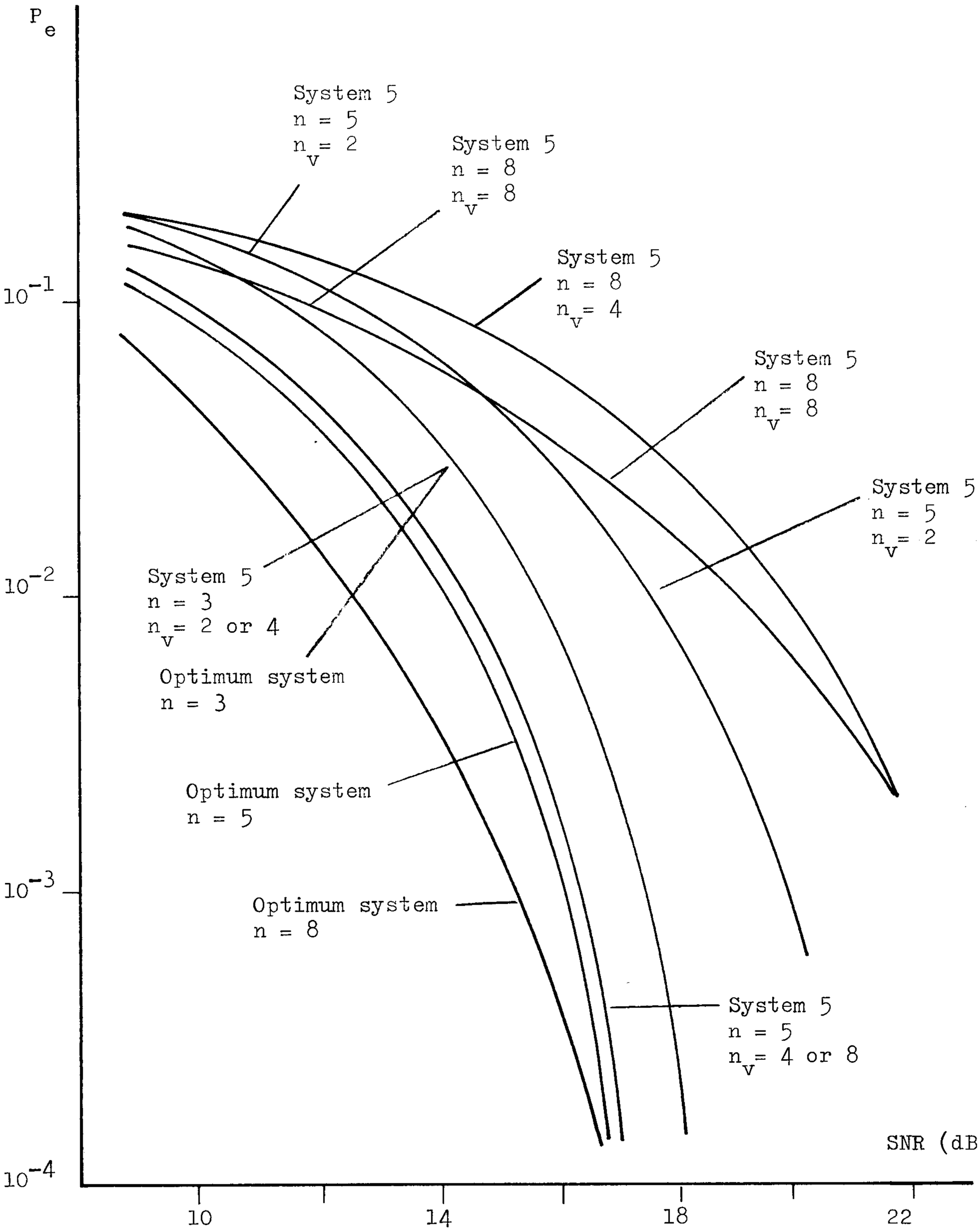


Fig. 4.12 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 5 operating over channel A (Table 2.1).

Number of filters used in the system (f) = 1 (Fig. 4.2).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

4.10 System 6

Only one filter in Fig. 4.2 is used in system 6, and it is selected to be the unit vector Z_1 (that is, $Z_1 Z_1^T = 1$) that maximizes the quantity

$$d = (Y_1 Z_1^T)^2 - \sum_{j=2}^{n} (Y_j Z_1^T)^2 \qquad (4.36)$$

where Y_j is the n-component vector defined by eqn. 3.9, and Z_1^T is the transpose of the row matrix Z_1, bearing in mind that a vector is also treated as a row matrix in this thesis. It can be seen from eqns. 4.5, 4.6, and 4.8 that, $Y_1 Z_1^T$, $Y_2 Z_1^T$, $\cdots$, $Y_n Z_1^T$ are the components associated with s_1, s_2, $\cdots$, s_n respectively in e_1, where e_1 is the quantity obtained at the output of the filter Z_1^T shown in Fig. 4.2. Consequently, in the detection of s_1 from R', $Y_1 Z_1^T$ may be considered as the signal component whereas $Y_2 Z_1^T$, $Y_3 Z_1^T$, $\cdots$, $Y_n Z_1^T$ may be considered as the intersymbol interference components in e_1. Eqn. 4.36 therefore suggests that, the filter Z_1^T used in system 6 effectively maximizes the relative magnitude of the energy of the signal component to the sum of the energy of the individual intersymbol interference component in e_1, for a given noise level in e_1, bearing in mind that the noise variance remains unchanged at the output of the filter here since $Z_1 Z_1^T = 1$ here.(A9) This is in fact the same filter as that used in any of the systems 9 - 11 (with $m_o = 1$) in chapter 3. Thus, eqn. 4.36 may be rearranged to become

$$d = Z_1 (A) Z_1^T \qquad (4.37)$$

and

$$A = Y_1^T Y_1 - \sum_{j=2}^{n} Y_j^T Y_j \qquad (4.38)$$

where A is here a $n \times n$ symmetric matrix. The Lagrange multipliers

used in system 5 is now used here to derive the vector Z_1 that maximizes the quantity d given by eqns. 4.37 and 4.38, with the constraint that $Z_1 Z_1^T = 1$ which ensures Z_1 to be an unit vector. The results obtained here are similar to those obtained in system 5 and are such that, the vector Z_1 selected for system 6 is the eigenvector associated with the largest eigenvalue of the matrix A given by eqn. 4.38, and the value of d defined by eqn. 4.36 is now equal to this eigenvalue.

It has been found, by using eqn. 4.8 to calculate and compare the values of $\{e_1'\}$, that there are altogether 2^n different possible values of $\{e_1'\}$ corresponding to the 2^n possible vectors $\{S\}$, for system 6 operating over channel A (Table 2.1). The value of n_m (maximum number of possible vectors $\{S\}$ having the same value of e_1') here is therefore equal to 1, and so there is effectively no constraint on the value of n_v (number of possible vectors $\{S\}$ selected at the output of the threshold device shown in Fig. 4.2) to be used in system 6 here, bearing in mind that the value of n_v is actually constrained to be such that $n_v \geq n_m$ (eqn. 4.9).

Computer simulation tests have been carried out to determine the tolerance to noise of system 6 operating over channel A whose sampled impulse response is as given in Table 2.1. The results of these tests are shown in Fig. 4.13 where the 95% confidence limit for each curve is about ± 0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 3.39 and 3.40 respectively. Fig. 4.13 also shows the performance of the optimum system described in section 4.3.

It can be seen from Figs. 4.4 - 4.13 that, when n = 8, f = 1, and $n_v = 8$, system 6 appears to have the best tolerance to noise over channel A (Table 2.1) of all the systems developed and studied here,

where f is the number of filters $\{Z_h^T\}$ used in the detection process (Fig. 4.2) of the given system. Nevertheless, the tolerance to noise of system 6 with n = 8, n_v = 8 or 4 is still very much poorer than that of the optimum system with n = 8. The optimum performance for n = 3 is achieved by system 6 with n_v = 2. When n = 5, the tolerance to noise of system 6 with n_v = 4 is about 0.5 dB lower than that of the corresponding optimum system at an error rate of 10^{-3}. The best performance of system 6 here appears to be achieved when the system operates with n = 5 and n_v = 8, and it approaches the optimum performance for n = 5. System 6 with a moderate value of n (being equal to 5 here) therefore appears to be very promising when operating over channel A here.

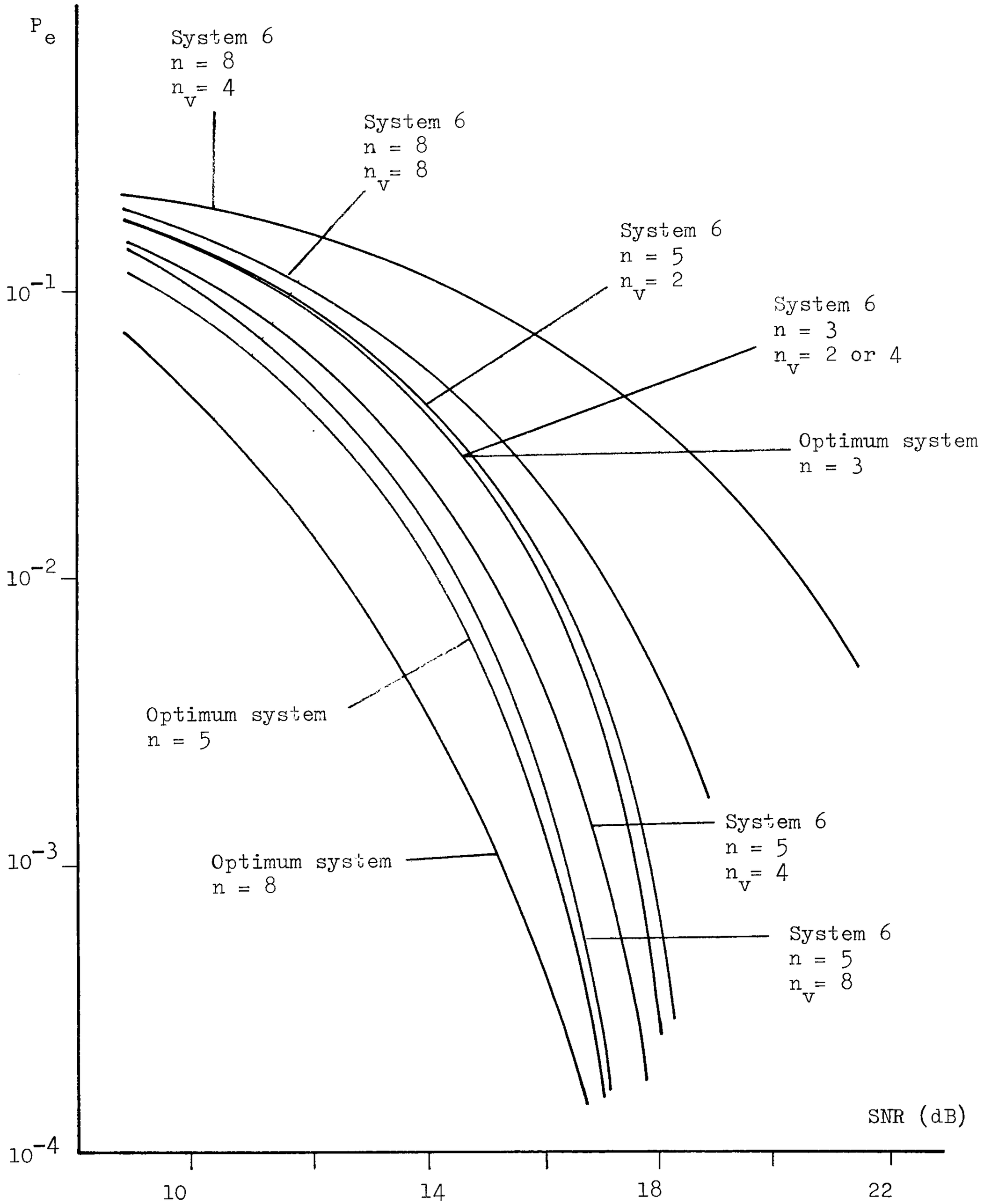


Fig. 4.13 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 6 operating over channel A (Table 2.1).

Number of filters used in the system (f) = 1 (Fig. 4.2).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

4.11 Further Computer Simulation Results

The tolerances to additive white Gaussian noise of the various systems developed in this chapter have so far been studied only over channel A (Table 2.1) which introduces severe pure amplitude distortion. Further computer simulation tests are now carried out to determine the tolerances to noise of the more promising systems here operating over channels B and C whose sampled impulse responses are as given in Table 2.1. The characteristics of these two channels have been described in section 2.4 and are such that, channel B introduces a combination of both phase distortion and amplitude distortion, whereas channel C introduces very severe amplitude distortion and no phase distortion. In the computer simulation tests here, 10,000 data symbols $\{s_i\}$ are transmitted for each measurement of the bit error rate, and the results are plotted as bit error rate P_e versus signal to noise ratio SNR. The definitions of P_e and SNR here are as given by eqns. 3.39 and 3.40 respectively. When the simulation results appear to be more scattered which normally occur at low error rates ($< 10^{-3}$), more simulation tests are carried out to measure the bit error rates at the same signal to noise ratios here. The results of these tests are shown in Figs. 4.14 - 4.18 for channel B and in Figs. 4.19 - 4.23 for channel C. The 95% confidence limits of these results are about ± 0.5 dB. The performances of the optimum system described in section 4.3 are also included here.

All computer simulation tests have been carried out by using the Prime 400 computer at the Loughborough University of Technology. The computer programs here are written in FORTRAN.

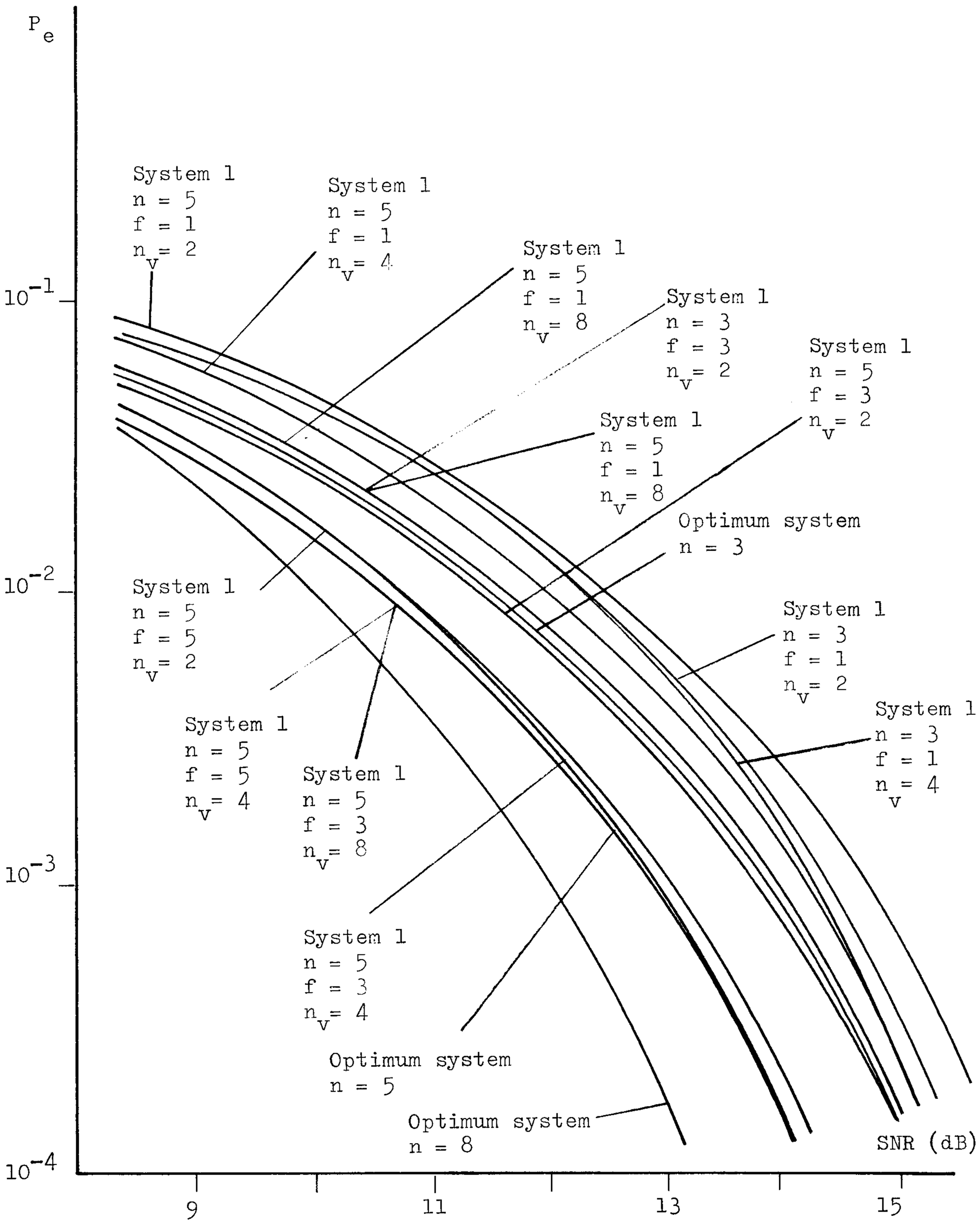


Fig. 4.14 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 1 operating over channel B (Table 2.1).

n : number of samples used in the system.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

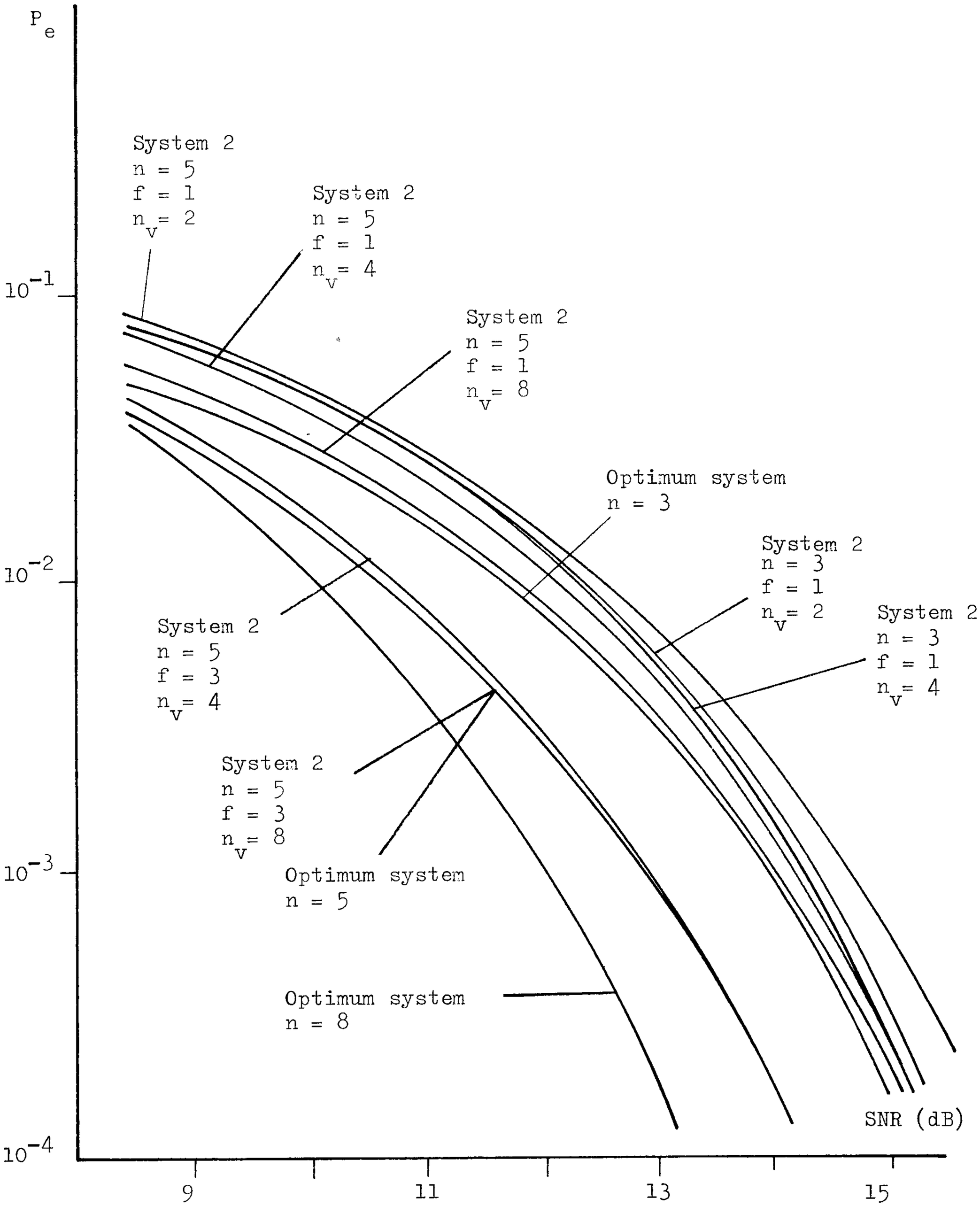


Fig. 4.15 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 2 operating over channel B (Table 2.1).

n : number of samples used in the system.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

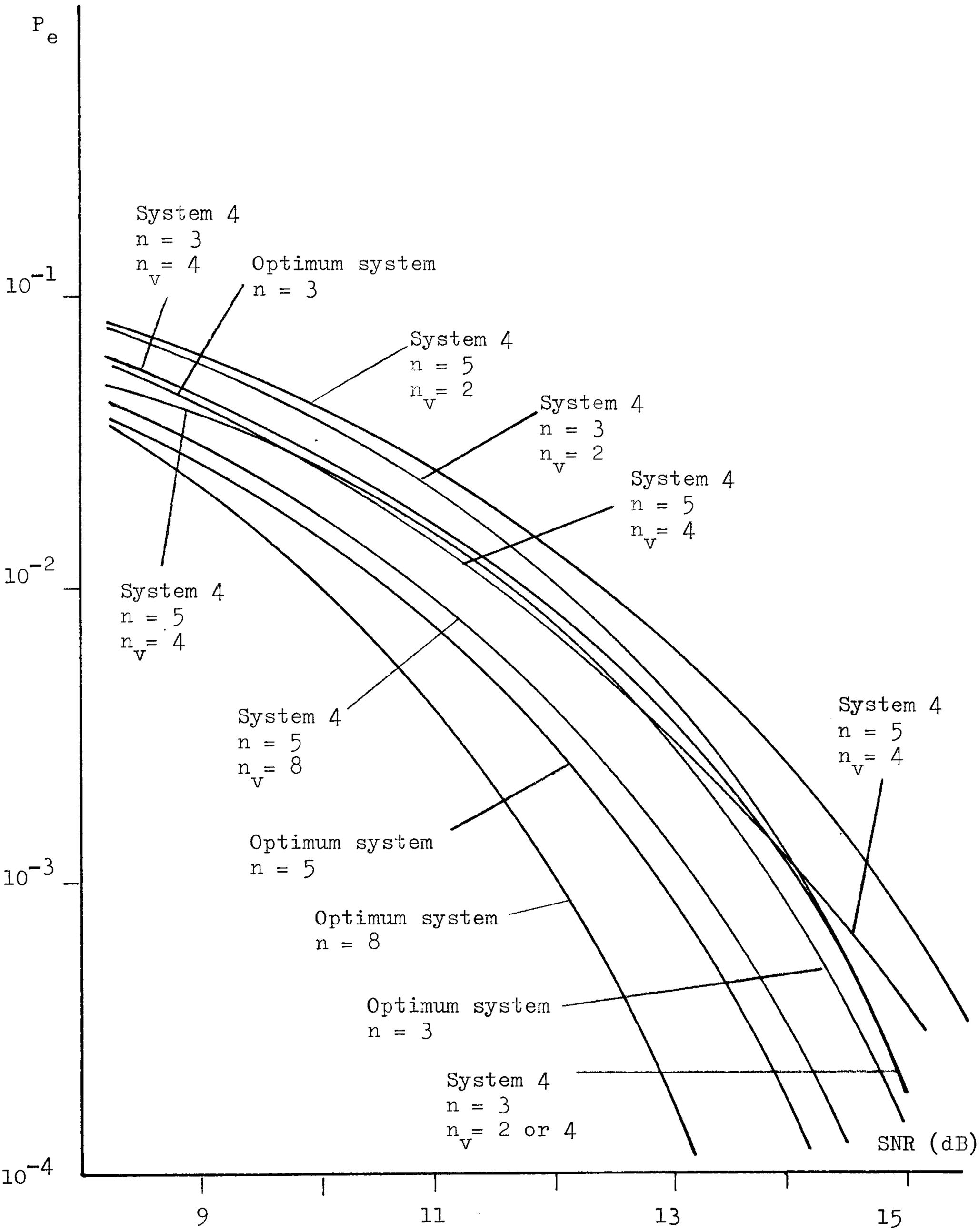


Fig. 4.16 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 4 operating over channel B (Table 2.1).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

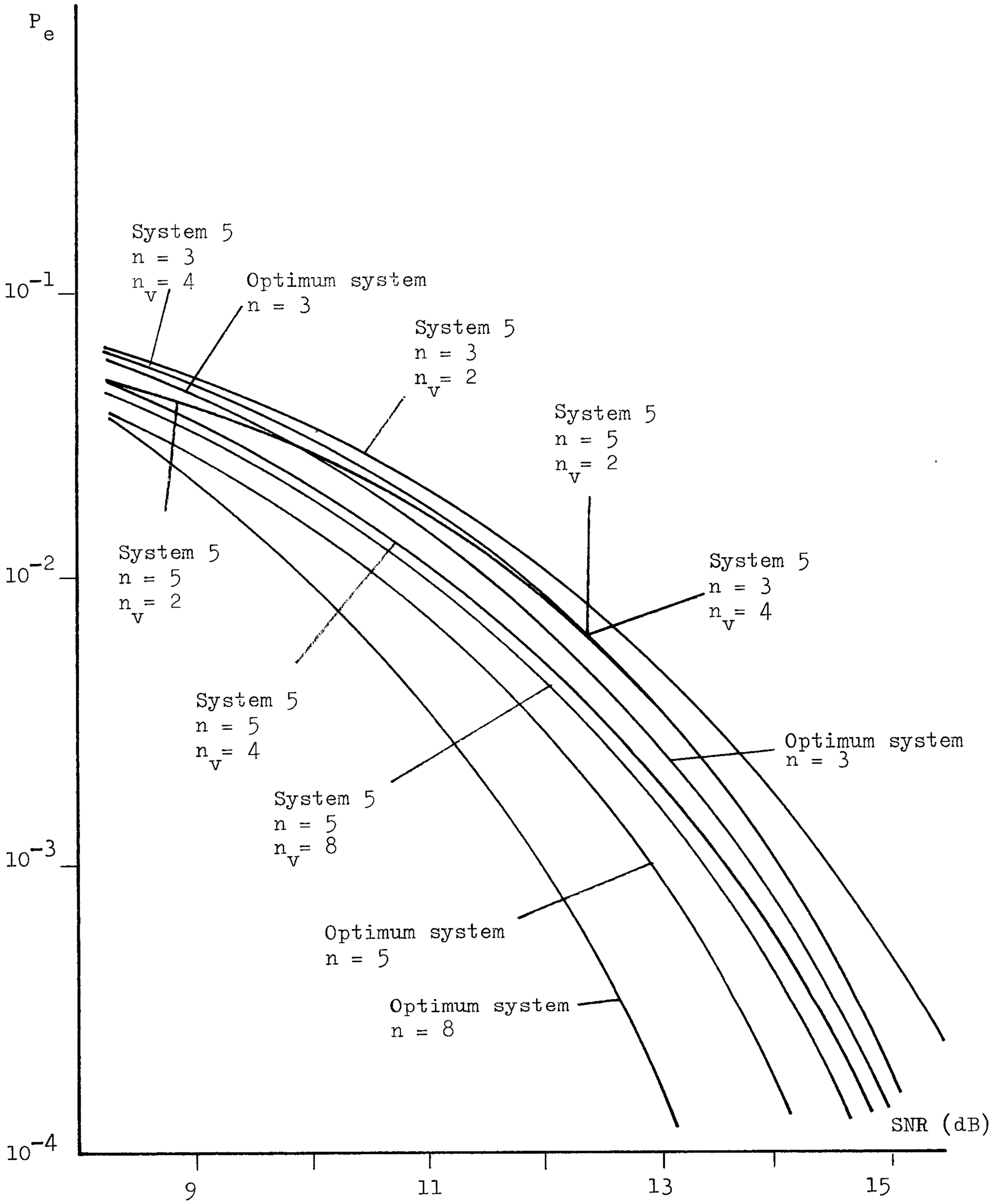


Fig. 4.17 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 5 operating over channel B (Table 2.1).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

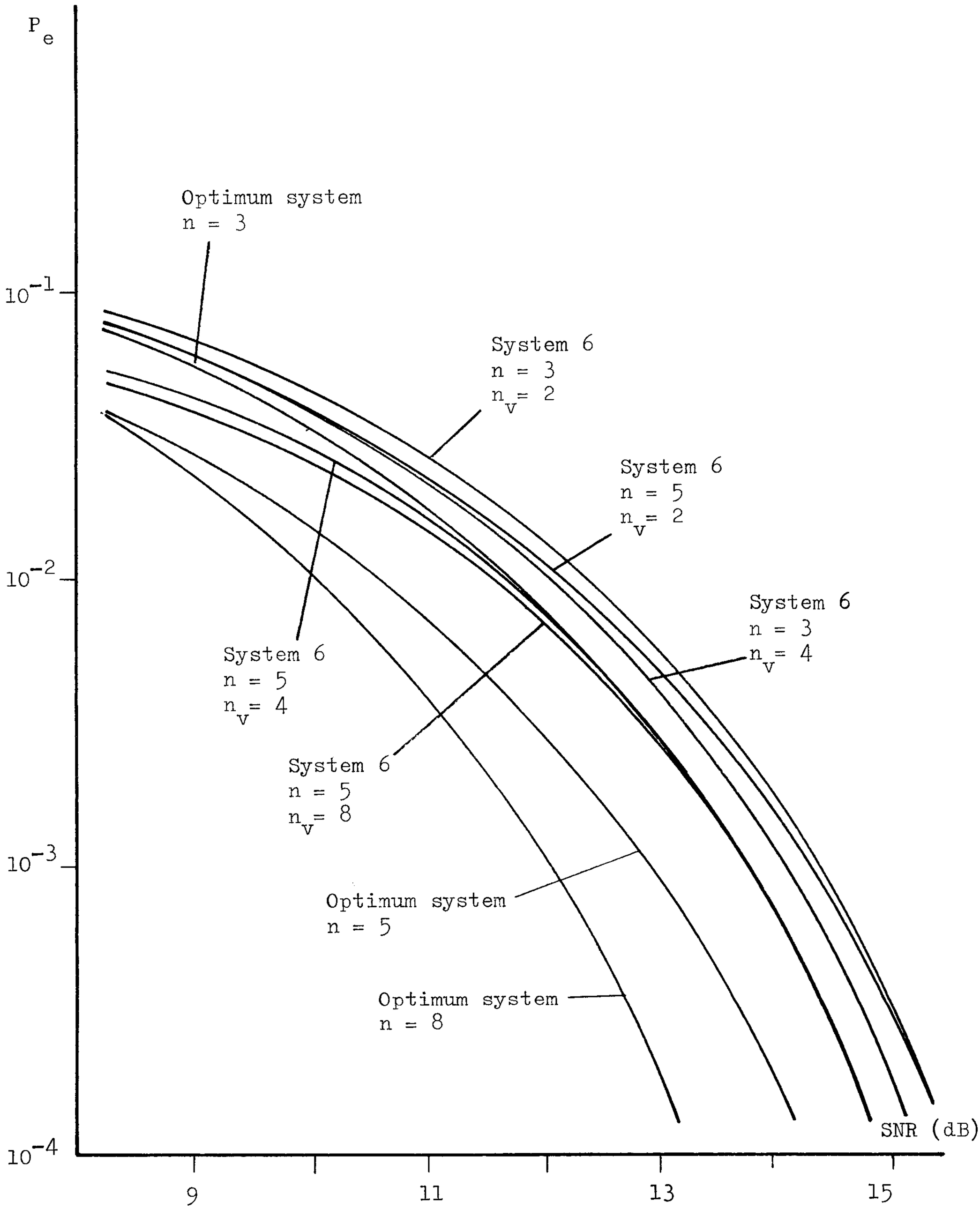


Fig. 4.18 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 6 operating over channel B (Table 2.1).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

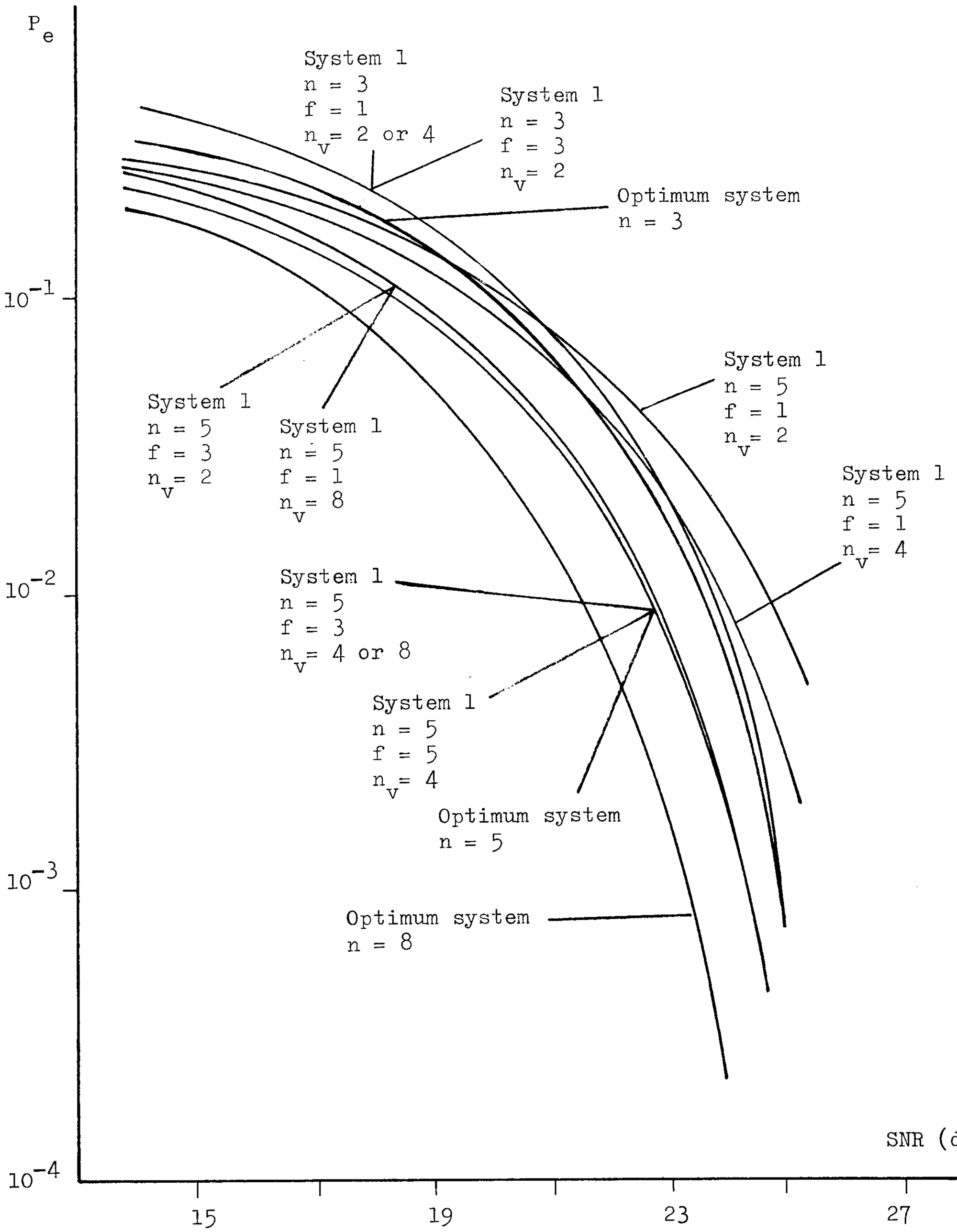


Fig. 4.19 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 1 operating over channel C (Table 2.1).

n : number of samples used in the system.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

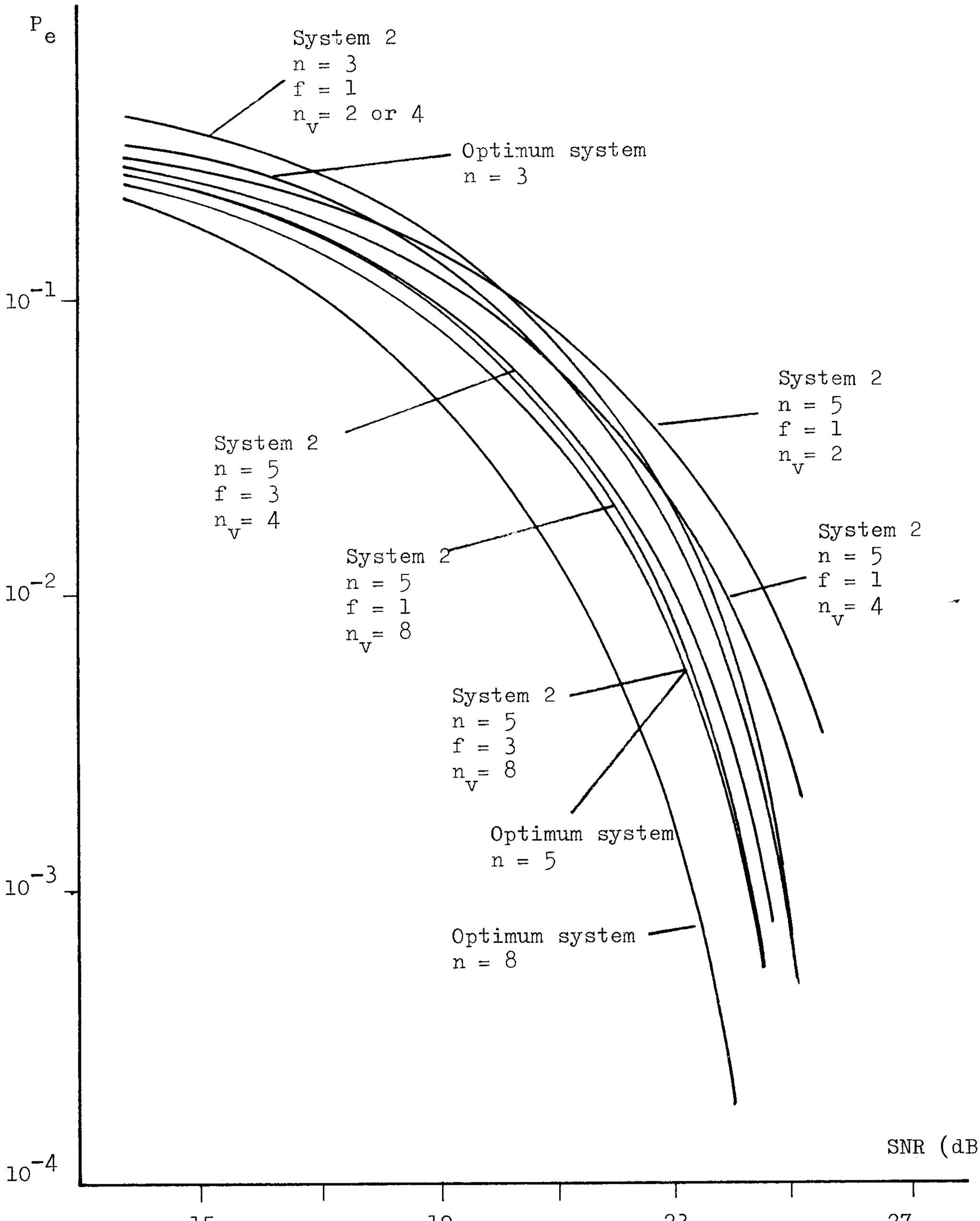


Fig. 4.20 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 2 operating over channel C (Table 2.1).

n : number of samples used in the system.

f : number of filters shown in Fig. 4.2.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

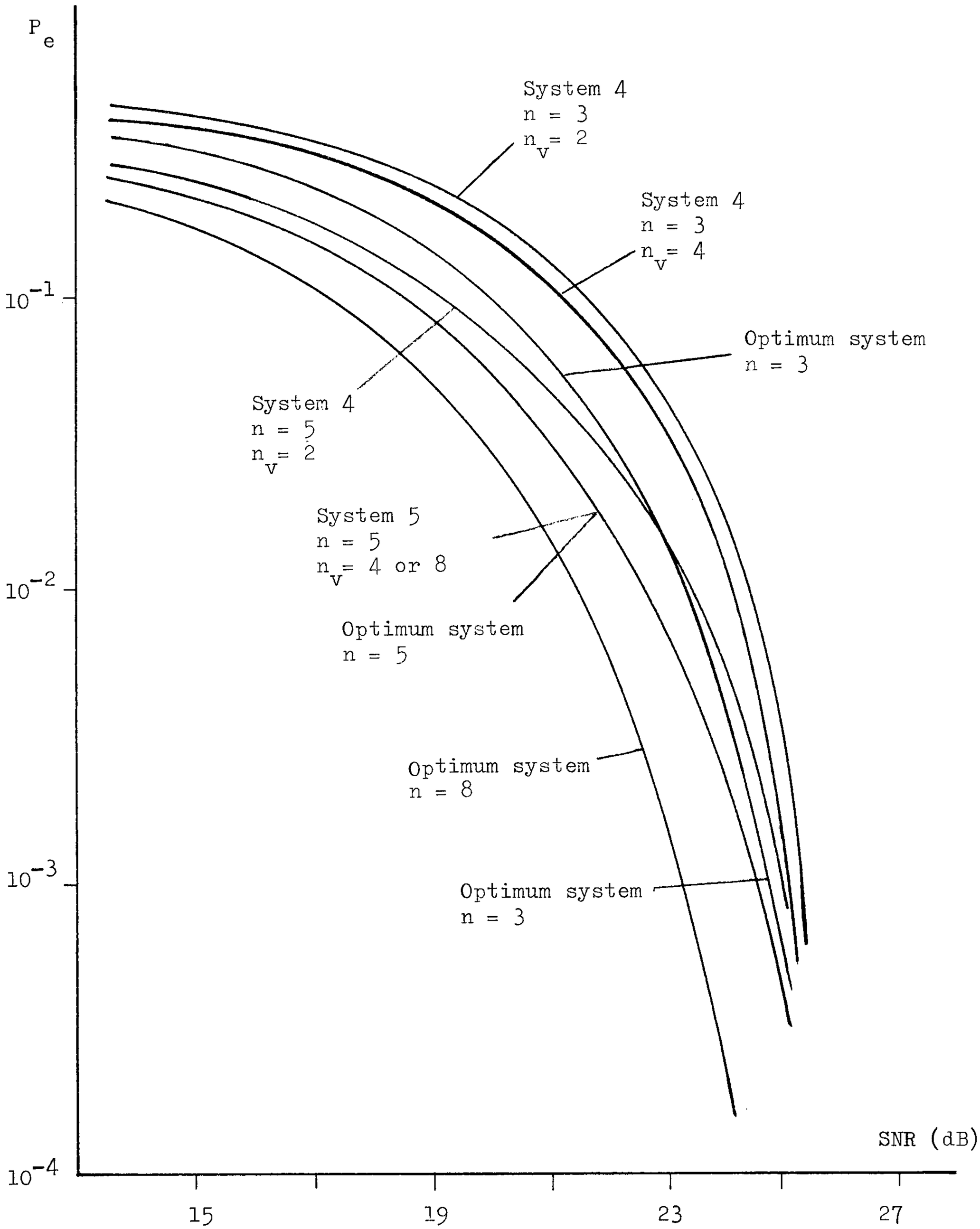


Fig. 4.21 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 4 operating over channel C (Table 2.1).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

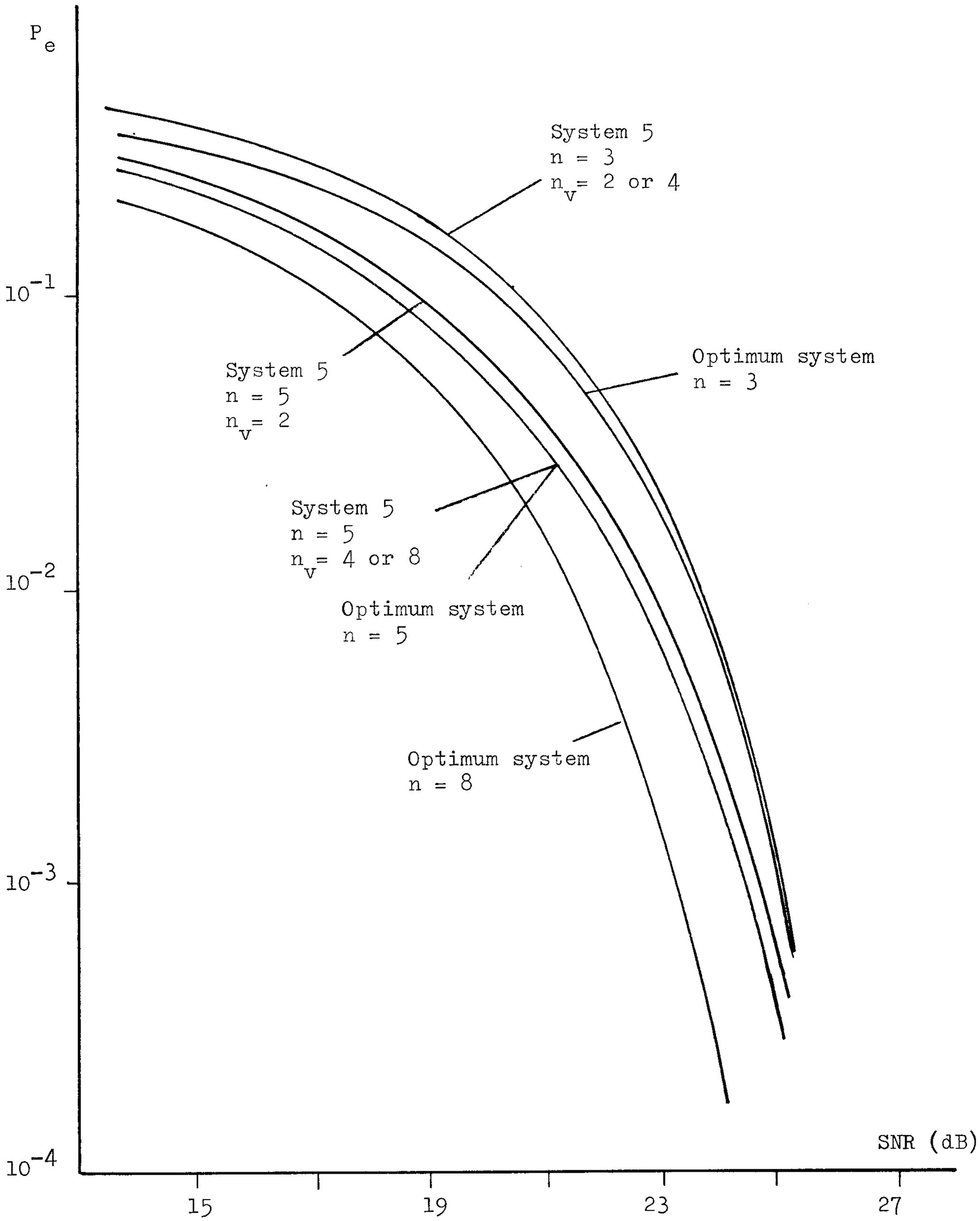


Fig. 4.22 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 5 operating over channel C (Table 2.1).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

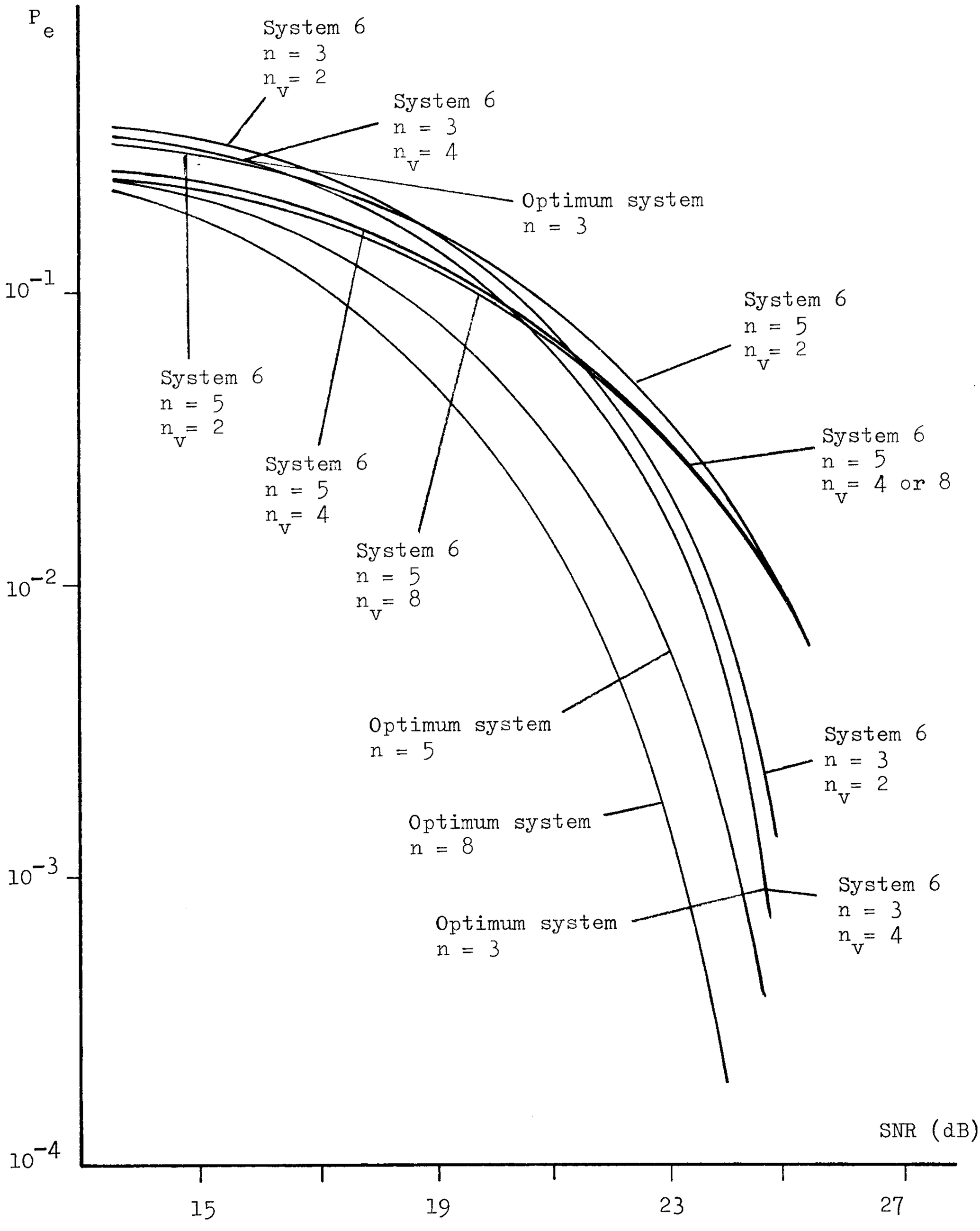


Fig. 4.23 Variation of error rate P_e (eqn. 3.39) with signal to noise ratio SNR (eqn. 3.40) for system 6 operating over channel C (Table 2.1).

n : number of samples used in the system.

n_v : number of $\{S\}$ selected in one threshold device (Fig. 4.2).

4.12 Assessment of Systems

All the systems developed in this chapter employ the same detection process shown in Figs. 4.1 and 4.2, and they differ only in having a different set of filter(s) $\{Z_h^T\}$ for the arrangement shown in Fig. 4.2. The selections of these filters for the more promising systems here are summarised as below.

System 1 : The filters here are selected to be matched to the vectors $\{Y_h\}$ defined by eqn. 3.9, with the constraint that $Z_h Z_h^T = 1$, for $h = 1, 2, \cdots, f$. f filters are used here and $1 \leq f \leq n$.

System 2 : The filters here are given as a set of orthonormal vectors which are derived from the $\{Y_h\}$ by using the Gram-Schmidt orthogonalisation process. f filters are used here and $1 \leq f \leq n$.

System 4 : Only one filter is used in this system and is given as the unit vector that makes the same acute angle with all the vectors $Y_1, Y_2, \cdots, Y_n$.

System 5 : Only one filter is used in this system and is selected to maximize the energy of the sampled impulse response of the channel and filter, with the constraint that $Z_1 Z_1^T = 1$.

System 6 : Only one filter is used in this system and is selected to maximize, in its output signal e_1, the relative magnitude of the energy of the signal component to the sum of the energy of the individual intersymbol interference component, with the constraint that $Z_1 Z_1^T = 1$.

In each of the systems 1 - 6, a total of (fn_v+2) distance measurements are involved in the detection of S from R', using the arrangement shown in Fig. 4.2, where n_v is the number of possible vectors $\{S\}$ selected at the output of each of the f threshold devices shown here. A distance is here referred to as the quantity $\|R' - SY\|$. In practice, the quantity $\|R' - SY\|^2$ will be evaluated instead of $\|R' - SY\|$, which simplifies the operations involved in the detection process without affecting the results.(A9) It should perhaps also be reminded that, in the detection of S from R', the number of distance measurements required in the optimum system described in section 4.3 is 2^n which is also the total number of possible vectors $\{S\}$. It is, of course, desirable to involve a small number of distance measurements in the detection process so that the corresponding system can operate at a high speed. Consequently, systems 1 - 6 are here compared with the optimum system in terms of the performance in tolerance to additive white Gaussian noise and the number of distance measurements required to achieve this performance.

Figs. 4.4 - 4.13 show the tolerances to noise of systems 1 - 6 operating over channel A. It can be seen from these results and the discussions given in the previous sections that, each of the systems 1 - 6 appears to have very poor tolerance to noise here when the given system operates with n = 8, f = 1, and n_v = 8 or 4 or 2. The number of distance measurements involved here with n_v = 8 is (fn_v+2) or 10 and is a very small number as compared to 256 which is the number of distance measurements required in the optimum system with n = 8. When n = 3, the optimum performance here is achieved by any of the systems 1, 2, 5, and 6 with f = 1 and n_v = 2. System 4 with the same values of f and n_v, however, has a tolerance to noise of about 1 dB lower than

that of the optimum system here (that is, when n = 3) at an error rate of 10^{-3}. When n = 5, the optimum performance here is achieved by any of the systems 1 and 2 with f = 3 and n_v = 8. However, the number of distance measurements involved here now becomes 26 and is about the same as that (2^n or 32) involved in the optimum system with n = 5. Each of the systems 1, 2, 4, 5, and 6 operating over channel A appears to have the best arrangement when the given system operates with n = 5, f = 1, and n_v = 8. The tolerance to noise here approaches to that of the optimum system with n = 5, and the number of distance measurements required to achieve this performance is only 10 which is obviously a small number as compared to that required in the optimum system here.

Figs. 4.14 - 4.18 show the tolerances to noise of systems 1, 2, 4, 5, and 6 operating over channel B. As it appears, when n = 5 and f = 1, system 4 with n_v = 8 appears to have the best tolerance to noise, with a loss of only about 0.5 dB in relation to that of the optimum system at an error rate of 10^{-3}. However, as n_v is reduced to 4, the tolerance to noise of system 4 degrades significantly, with a further loss of about 2 dB at the same error rate. System 5 therefore appears to be the best of all the systems here, as its tolerance to noise is only about 0.5 dB inferior to that of system 4 at n_v = 8, but is about 1 dB better than the latter at n_v = 4. Each of the systems 1, 2, and 6 has about the same tolerance to noise here and is a little inferior to that of system 4 or 5. When n = 3 and f = 1, roughly the same tolerance to noise is achieved by any of the systems 4 and 5 with n_v = 4 or 2, and is about 0.5 dB inferior to that of the optimum system (with n = 3) at an error rate of 10^{-3}. The performance of any of the systems 1, 2, and 6 is here slightly inferior to that of any of the systems 4 and 5.

Figs. 4.19 - 4.23 show the tolerances to noise of systems 1, 2, 4, 5, and 6 operating over channel C. It can be seen from these results that, when n = 5 and f = 1, each of the systems 4 and 5 appears to have achieved the optimum performance with $n_v = 4$, and each of the systems 1 and 2 appears to have achieved it with $n_v = 8$. System 6, however, appears to be the poorest of all the systems here, and with $n_v = 8$, it gives a tolerance to noise of about 2 dB lower than that of the optimum system (with n = 5) at an error rate of 10^{-3}. When n = 3 and f = 1, each of the systems 1, 2, 4, 5, and 6 with $n_v = 4$ or 2 is seen to achieve roughly the same performance which approaches to that of the optimum system here at error rates below 10^{-3}.

It can now be seen from the results just described that, given the values of n, f, and n_v, systems 1, 2, 4, 5, and 6 appear to have about the same tolerance to noise, with a difference of not more than 2 dB (and very often much less than 2 dB) at an error rate of 10^{-3}, when operating over any of the channels A, B, and C. One weakness of any of these systems appears to be that, when the given system operates with a large value of n, a large number of possible vectors $\{S\}$ are required to be selected and hence a large number of distance measurements are required to be evaluated in the system in order to achieve a given tolerance to noise. This is probably because the total number of possible vectors $\{S\}$ is given by 2^n so that, for a large value of n, a large number of possible vectors $\{S\}$ are associated with small distances and so a large number of these vectors are required to be selected in the system here so as to include the vector associated with the minimum distance, bearing in mind that the optimum system here selects as the detected vector the possible vector of S associated with the minimum distance. As the results in Figs. 4.4 - 4.13 show, when n = 8, each

of the systems 1 - 6 with f = 1 and n_v = 8, 4, or 2 appears to have very poor tolerance to noise as compared to that of the optimum system. This value of n therefore appears to be too large for the values of n_v considered here, bearing in mind that each of the systems 3 - 6 operates only with f = 1 and that (fn_v+2) is the number of distance measurements involved in the system here. The weakness of using a small value of n in the system here is that, the optimum performance achievable by the system may be degraded to such a large extent that it is now impossible to achieve a useful tolerance to noise by the system. As Figs. 4.4 - 4.23 show, for the three channels A, B, and C (Table 2.1) tested here, the tolerances to noise of the optimum system with n = 8 are about 0.5 to 1.5 dB better than that of the optimum system with n = 5 and are about 2 to 2.5 dB better than that of the optimum system with n = 3, at an error rate of 10^{-3}. Consequently, with n = 3, none of the systems 1 - 6 can achieve a tolerance to noise near to that of the optimum system with n = 8. The best arrangement of each of the systems 1, 2, 4, 5, and 6 therefore appears to be that when a moderate value of n is used in the system. As can be seen from the results shown in Figs. 4.4 - 4.23, each of the systems 1, 2, 4, 5, and 6 with n = 5 operating over any of the channels A, B, and C, is able to achieve a good performance with a small number of operations. In particular, when system 5 operates with n = 5, f = 1, and n_v = 4, the tolerance to noise achieved here is about 0.5 dB inferior to that of the optimum system with n = 5 at an error rate of 10^{-3}, for each of the channels A, B, and C. The number of distance measurements required in this arrangement of system 5 is only (fn_v+2) or 6 which is obviously a very small number as compared to that (2^n or 32) of the optimum system with n = 5.

Thus, the detection process with any of the arrangements of systems 1, 2, 4, 5, and 6 developed in this chapter appears to be very promising when a moderate value of n (being equal to 5 for the three channels tested here) is used in the system. The operation of the more promising arrangement of this detection process is very simple and it involves only the linear filtering of a set of n sample values, the selection of a set of vectors by using a simple multi-level threshold device, and the measurement of a distance for each of these selected vectors. This detection process is also able to achieve a very satisfactory tolerance to additive white Gaussian noise with a small number of distance measurements. The channel sampled impulse response of the data-transmission system here is, however, restricted to be known and time-invariant so that the parameters of the filter and the threshold values used in the detection process here can be evaluated before the transmission of data begins.

CHAPTER 5

NEAR-MAXIMUM LIKELIHOOD DETECTION PROCESSES FOR A 16-POINT QAM SIGNAL TRANSMITTED OVER A TELEPHONE CIRCUIT

5.1 Introduction

In a digital data-transmission system operating over the switched telephone network, it is essential to employ an adaptive receiver that can be used over any telephone circuit which may have any one of a wide range of characteristics. The adaptive receiver here may be implemented by using an adaptive nonlinear (decision feedback) equalizer.(B16,B22,B34,B36) Such systems have been known to operate satisfactorily at transmission rates up to 4800 bits/second, above which satisfactory performance is not always achievable over the poorer telephone circuits.(C42)

Alternatively, the adaptive receiver may be implemented by using the Viterbi-algorithm detector described in section 2.7.2. It is known that, under the various conditions assumed in section 2.7, the Viterbi-algorithm detector is able to achieve the same performance as that of the maximum-likelihood detector which minimizes the probability of error in the detection of the whole transmitted message.(A9) Unfortunately, the amount of storage and the number of operations per data symbol involved in the Viterbi-algorithm detector increases exponentially with the number of non-zero components of the channel sampled impulse response. In the particular application considered in this chapter, where the sampled impulse response of the telephone channel contains a large number of non-zero components, direct application of the Viterbi-algorithm detector becomes too complex and impractical.

One approach for overcoming this problem is to insert a linear filter at the input to the Viterbi-algorithm detector, so that the overall sampled impulse response of the channel and filter now has a small number of components.[C13,C16,C20,C33] However, this linear filter may equalize some of the amplitude distortion introduced by the telephone circuit, so that the true maximum-likelihood detection is no longer achieved, leading to inferior performance.[A9] The linear filter here should ideally perform the function of a 'whitened matched-filter';[C10,C42] which does not remove any of the amplitude distortion introduced by the telephone circuit. A Viterbi-algorithm detector operating at the output of this filter now has the same tolerance to Gaussian noise as the appropriate Viterbi-algorithm detector operating at its input, although generally involving much less complex equipment.[C42] However, since this filter does not remove any of the amplitude distortion introduced by the telephone circuit, the sampled impulse response of the channel and filter does not always, or even in general, have a sufficiently small number of components to permit the practical use of the Viterbi-algorithm detector. In the particular application considered in this chapter, where a 16-point QAM signal is transmitted over a telephone circuit at 9600 bits/second, practical implementation of the Viterbi-algorithm detector can become unduly complex even if the number of components in the sampled impulse response of the channel and filter is as few as three. It therefore follows that, the Viterbi-algorithm detector must now be replaced by a near-maximum likelihood detector that uses very much less storage and requires far fewer operations in the detection process.[C37,C38,C42]

The weakness of the arrangement described above (which involves the use of a linear prefilter that acts as a 'whitened matched-filter')

is that, a fair amount of time is normally required to adjust the tap gains of the linear prefilter here (being implemented as a linear feedforward transversal filter) to their correct values, so that the detection process here now has a long starting up procedure. Furthermore, a large number of taps are usually required to be used in this prefilter so as to achieve the 'desired' sampled impulse response of the channel and prefilter here with an acceptable accuracy. This means that a large number of multiplication processes are now required to be carried out in this prefilter.

It appears that two approaches may be used to overcome some or all of the weaknesses mentioned above. One approach is to use a simpler linear prefilter which has far fewer taps, the tap-gain values here being easily evaluated from the channel sampled impulse response. The second approach is to discard entirely the use of a prefilter, and to use just a more efficient and probably therefore more complex near-maximum likelihood detector for the detection process. Various arrangements of these two types of detection processes have been developed and studied in this chapter.

5.2 Basic Assumptions

The data-transmission system here is the synchronous serial 16-point QAM data-transmission system described in section 2.4, where the signals are transmitted at 9600 bits/second over a telephone circuit. The equivalent baseband model of this data-transmission system is shown in Fig. 2.3. The signal processor shown in Fig. 2.3 is here implemented as a near-maximum likelihood detection process which may or may not include the use of a linear prefilter at its front end. Thus, from eqn. 2.19, the

sample value of the received signal at the input to the signal processor shown in Fig. 2.3, at time t = iT, is here given by the complex-valued quantity

$$r_i = \sum_{h=0}^{g} s_{i-h} y_h + w_i \qquad (5.1)$$

where the $\{w_i\}$ are the complex-valued noise components, $y_0, y_1, \cdots, y_g$ are the g+1 complex-valued components of the channel sampled impulse response, and the $\{s_i\}$ are the 16-point QAM data symbols whose values are to be determined here. It is assumed that the real and imaginary parts of the $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and variance σ^2, where $2\sigma^2$ is the two-sided power spectral density of the noise added at the output of the transmission path shown in Fig. 2.3. It is also assumed that the $\{s_i\}$ are statistically independent and equally likely to have any of their 16 possible values, where

$$s_i = a_i + jb_i \qquad (5.2)$$

and $j = \sqrt{-1}$, $a_i = \pm 1, \pm 3$, $b_i = \pm 1, \pm 3$. The signal processor (Fig. 2.3) operates on the sample values $\{r_i\}$ given by eqn. 5.1, to give at its output the detected values of the data symbols $\{s_i\}$. The detected value of s_i is here designated as s_i'.

Various arrangements of the signal processor that uses a near-maximum likelihood detector are studied in this chapter. The arrangement where a simple linear prefilter is inserted ahead of a simple near-maximum likelihood detector is first considered. The investigation then continues with the studies of the arrangement where no prefilter is used in the signal processor. The performance of these systems are compared with those of the optimum nonlinear equalizer described in section 2.6.3, and the system to be described in the next section where

a linear filter that acts as a 'whitened matched-filter' is inserted at the input of a simple near-maximum likelihood detector. In order to have a better understanding of the systems developed and studied here, the channel sampled impulse response is assumed to be known and time invariant for the whole transmission period.

The performances in tolerances to additive white Gaussian noise of the systems developed and studied in this chapter are evaluated by using computer simulation tests. In these tests, 40,000 binary digits are generated and converted (Table 2.2) into the corresponding 10,000 16-point QAM data symbols $\{s_i\}$ for transmission (assuming the data-transmission of Fig. 2.3), and the number of errors in the detection of these binary digits are recorded for each measurement of the bit error rate, bearing in mind that the same Gray code (Table 2.2) is used at the receiver to decode the detected data symbols $\{s_i'\}$ into the corresponding detected binary digits here. The results of the computer simulation tests here are plotted as bit error rate P_e versus signal to noise ratio SNR, where P_e is defined as

$$P_e = \frac{\text{Number of errors in the detection of binary digits}}{\text{Total number of binary digits transmitted}} \qquad (5.3)$$

When the simulation results appear to be more scattered which normally occur at low error rates ($< 10^{-3}$), more simulation tests are carried out to measure the bit error rates at the same signal to noise ratios here. The signal to noise ratio SNR here is defined as

$$\text{SNR} = 10 \log_{10} \left(\frac{E'}{\frac{1}{2}N_o} \right) \quad \text{dB} \qquad (5.4)$$

where E' is the average transmitted energy per data sybmol s_i, at the

input to the transmission path (Fig. 2.3), and N_o is the two-sided power spectral density of the additive white Gaussian noise at the input to the receiver filter. The channel sampled impulse responses to be used in the computer simulation tests here are those shown in Tables 2.4 - 2.7. In order to avoid using excessive computing time, only channel **A** (Tables 2.4 - 2.7) is used in the initial computer simulation tests to determine the tolerances to noise of the systems developed and studied in this chapter. The more promising systems here are then further tested over each of the channels B - F (Tables 2.4 - 2.7) to ensure a more representative test. All computer simulation tests here have been carried out on the **CDC** 7600 computer in Manchester.

5.3 System 1

This is the system referred to as system B in Ref. C42, and it is considered here as a reference system for comparison with the systems developed in this chapter. The basic model of the signal processor here is shown in Fig. 5.1.

In Fig. 5.1, the 'optimum' linear prefilter is the filter that acts as a 'whitened matched-filter' described previously, and it is basically an all-pass network such that the sampled impulse response of the channel and filter is minimum phase.(C42) This filter effectively replaces all zeros of the z-transform of the channel that lie outside the unit circle in the z-plane by the complex conjugates of their reciprocals, leaving the remainning zeros unchanged. Thus, the sampled impulse response of the channel and optimum prefilter here has g+1 components $y'_0, y'_1, \cdots, y'_g$ which have a z-transform with all the zeros lying inside or on the unit circle in the z-plane. The first

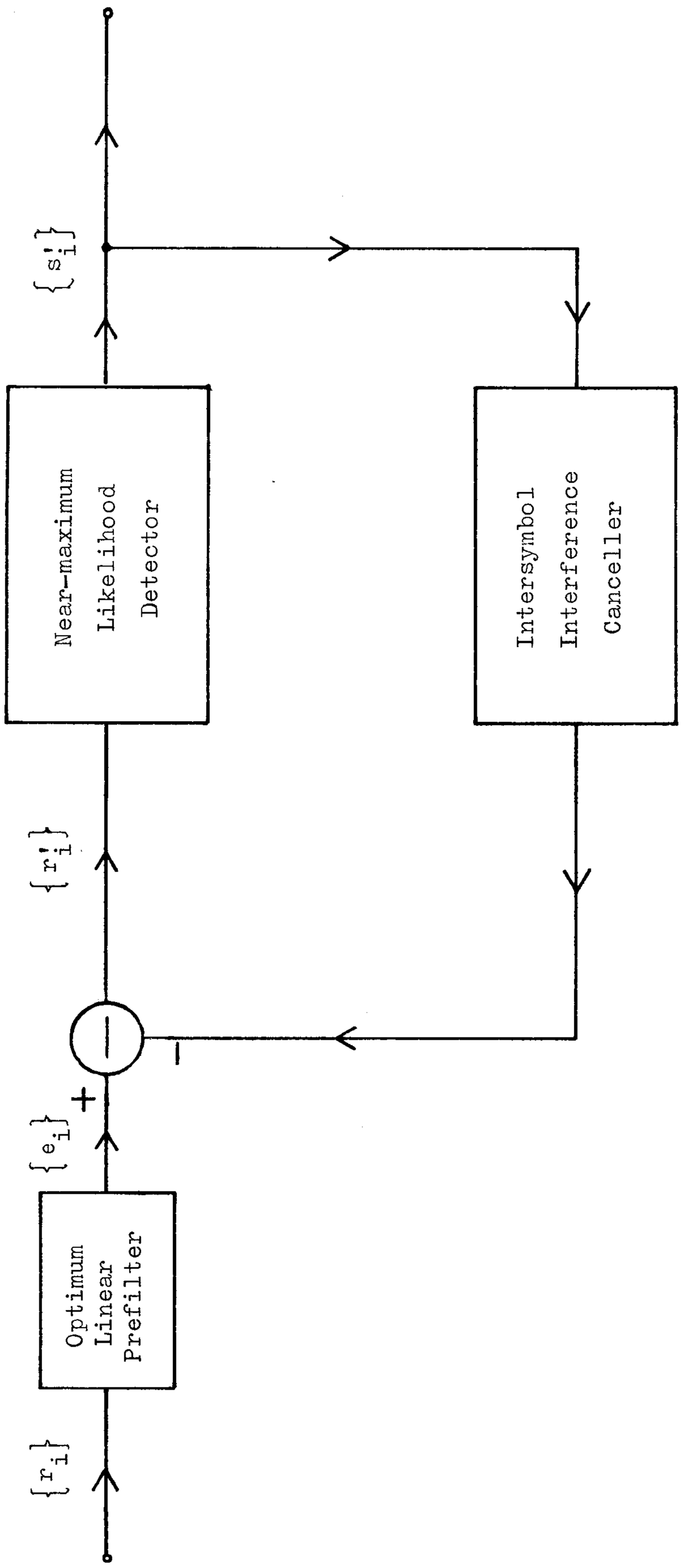


Fig. 5.1 Basic model of the signal processor for system 1

component y'_0 is here one of the larger components in the sampled impulse response of the channel and prefilter.(C42) Furthermore, the noise components at the output of this prefilter have the same statistical properties as the noise components at its input.(C42) Thus, the sample value at the output of the prefilter in Fig. 5.1, at time t = iT, is given by the complex-valued quantity

$$e_i = \sum_{h=0}^{g} s_{i-h} y'_h + w'_i \qquad (5.5)$$

where the $\{w'_i\}$ are the noise components whose real and imaginary parts are statistically independent Gaussian random variables with zero mean and variance σ^2, and the $\{s_i\}$ are the 16-point QAM data symbols (eqn. 5.2) whose values are to be determined here. The components $y'_0, y'_1, \cdots, y'_g$ in eqn. 5.5 are, of course, the g+1 components of the sampled impulse response of the channel and prefilter here.

The detector (Fig. 5.1) operates on its input samples $\{r'_i\}$ to give the finally detected data symbols $\{s'_i\}$, s'_i being determined after the receipt of r'_{i+n}, where $n < g$, so that there is a delay in detection of n sampling intervals. The operation of this detector will shortly be described. The intersymbol interference canceller (Fig. 5.1) removes from the samples $\{e_i\}$ (eqn. 5.5), detected values of all components involving data symbols $\{s_i\}$ whose final detected values $\{s'_i\}$ have already been determined. That is, the intersymbol interference canceller operates on e_i to give the complex-valued quantity

$$r'_i = e_i - \sum_{h=n+1}^{g} s'_{i-h} y'_h \qquad (5.6)$$

Assuming for the moment that $s'_{i-h} = s_{i-h}$ for $h = n+1, n+2, \cdots, g$,

eqn. 5.6 now reduces to

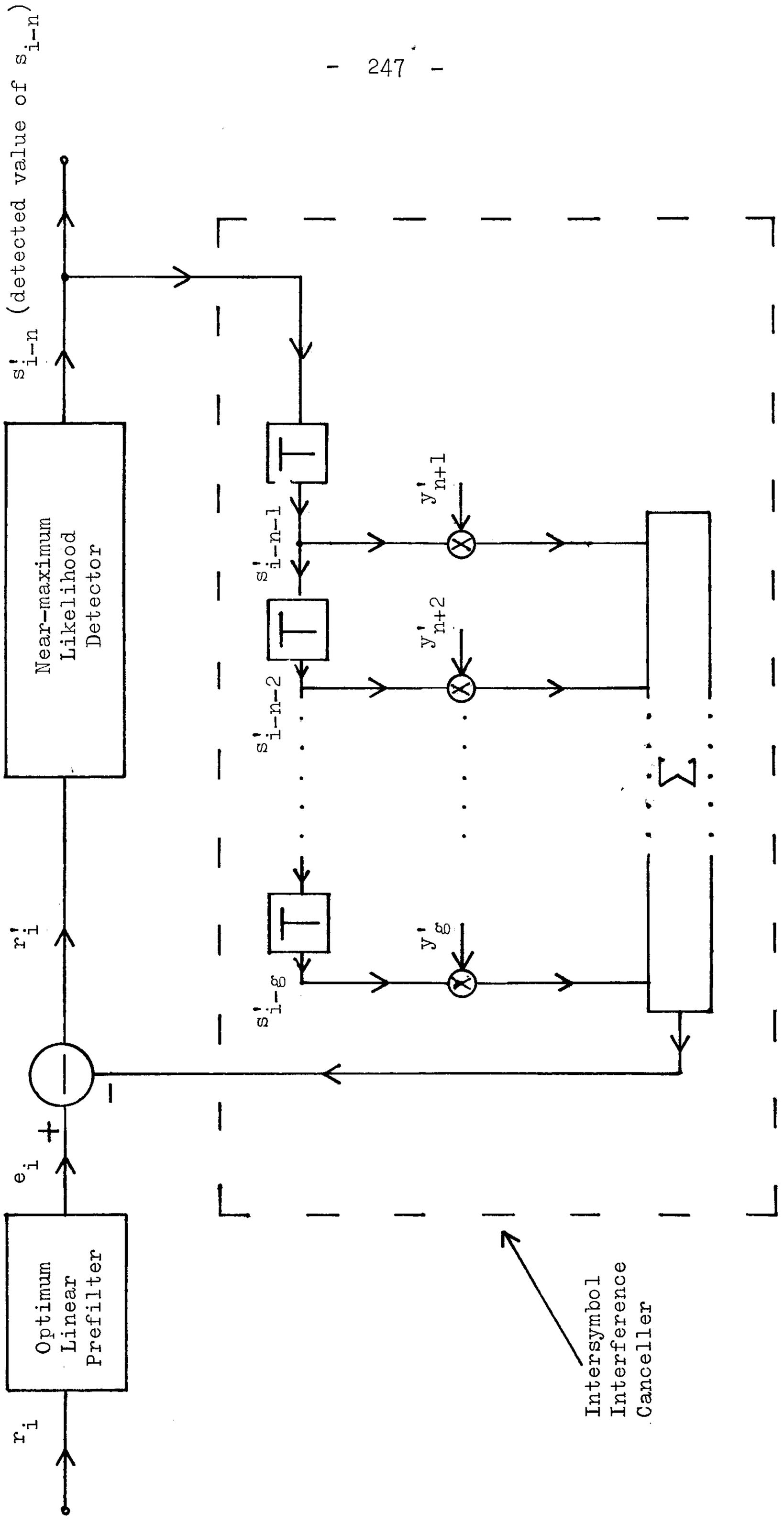
$$r_i' = \sum_{h=0}^{n} s_{i-h} y_h' + w_i' \qquad (5.7)$$

It can be seen from eqns. 5.5 and 5.7 that, the effective number of components of the sampled impulse response of the channel and prefilter has been reduced from g+1 to n+1, so that the detection process is now greatly simplified when $n < g$. Fig. 5.2 shows the arrangement for the intersymbol interference canceller just described. For the purpose of study here (where the sampled impulse response of the channel and prefilter is assumed to be known and time invariant), the arrangement of the adaptive adjustment of the prefilter and the intersymbol interference canceller for a time varying channel are not shown in Fig. 5.2. In Fig. 5.2, a square marked T is a device which introduces a delay of T seconds, where T seconds is the sampling interval, and a square marked $\sum$ is an accumulator that sums the input samples. Clearly, when there is no delay in detection (that is, n = 0), the intersymbol interference canceller shown in Fig. 5.2 becomes identical to that used in the pure nonlinear equalizer described in section 2.6.2.

The operation of the near-maximum likelihood detector shown in Fig. 5.2 for system 1 is now described as follows. Just prior to the receipt of the sample r_i' at the detector input, the detector holds in store m n-component vectors $\{X_{i-1}\}$ where

$$X_{i-1} = \begin{bmatrix} x_{i-n} & x_{i-n+1} & \cdots & x_{i-1} \end{bmatrix} \qquad (5.8)$$

and x_k has one of the 16 possible values of s_k for any positive integer value of k. Each vector X_{i-1} is associated with a cost C_{i-1} which will shortly be discussed.



Signals present immediately after the detection of s_{i-n} in the signal processor of system 1 (Fig. 5.1).

Fig. 5.2

On receiving r_i', each of the m stored vectors $\{X_{i-1}\}$ is expanded into 16 (n+1)-component vectors $\{P_i\}$, where

$$P_i = \begin{bmatrix} x_{i-n} & x_{i-n+1} & \cdots & x_i \end{bmatrix} \tag{5.9}$$

The first n components of P_i are as in the original vector X_{i-1}, and the last component x_i has the 16 possible values of s_i (given by a_i+jb_i for $a_i = \pm 1, \pm 3$ and $b_i = \pm 1, \pm 3$). The detector then evaluates the cost C_i for each of the 16m vectors $\{P_i\}$. The cost C_i is here defined as

$$C_i = c_1 + c_2 + \cdots + c_i \tag{5.10}$$

or

$$C_i = C_{i-1} + c_i \tag{5.11}$$

and the quantity c_k is here given by

$$c_k = |r_k' - q_k'|^2 \tag{5.12}$$

where

$$q_k' = \sum_{h=0}^{n} x_{k-h} y_h' \tag{5.13}$$

and r_k' is as defined by eqn. 5.6, for $k = 1, 2, \cdots, i$. The quantity $|r_k' - q_k'|$ in eqn. 5.12 is the absolute value (modulus) of $r_k' - q_k'$. Thus, if R_i' and Q_i' are the i-component vectors whose kth components are given by r_k' and q_k' respectively, then the cost C_i here is the square of the unitary distance between R_i' and Q_i'. Under the various conditions assumed here, the maximum likelihood vector $\begin{bmatrix} x_1 & x_2 & \cdots & x_i \end{bmatrix}$ is its possible vector such that C_i is minimized, and this vector is the possible vector of $\begin{bmatrix} s_1 & s_2 & \cdots & s_i \end{bmatrix}$ most likely to be correct.(C42) Thus, the cost C_i for each of the 16m expanded vectors $\{P_i\}$ is here evaluated by using eqns. 5.11, 5.12, and 5.13, bearing in mind that

the cost C_{i-1} in eqn. 5.11 has already been evaluated in the previous detection process. The detector now proceeds to select the vector of $\{P_i\}$ here associated with the smallest cost, and it takes the value of x_{i-n} in this vector to be the value of the detected data symbol s'_{i-n}. All vectors $\{P_i\}$ for which $x_{i-n} \neq s'_{i-n}$ are then discarded, and the first component (being x_{i-n}) of each of the remaining vectors $\{P_i\}$ is omitted to give the corresponding n-component vectors $\{X_i\}$. The cost of a vector X_i here is taken to be that of the corresponding vector P_i. Thus, the vector X_i associated with the smallest cost is here the first selected vector. The detector then selects the second vector X_i associated with the smallest cost subject to the constraint that only the 16 vectors $\{X_i\}$ deriving from the previous first (best) vector X_{i-1} are available for the selection here, bearing in mind that the first selected vector X_i is not to be selected again. This constraint is to ensure that the previous best vector is retained so that, in the case of severe impulsive noise where the noise samples are very large, not all the vectors deriving from the previous best vector are discarded from the store. Having selected the first two vectors, the detector then proceeds to select from the remaining vectors $\{X_i\}$ the m-2 vectors associated with the smallest costs, to give a total of m selected vectors $\{X_i\}$. These m selected vectors are stored together with their associated costs $\{C_i\}$. The detector is now ready for the next detection process where the data symbol s_{i-n+1} will be detected in the same way as is described above.

At the start of transmission, the n components $x_1, x_2, \cdots, x_n$ of each of the m stored vectors $\{X_n\}$ in the detector here are set to the corresponding values of the data symbols $s_1, s_2, \cdots, s_n$ whose values are known at the receiver. However, only the cost C_n of the

first stored vector is set to zero, whereas the costs of the remaining m-1 stored vectors here are set to some very high values. Consequently, after a few detection processes have been completed, all the m stored vectors $\{X_i\}$ $(i > n)$ will have been derived from the vector X_n associated with a zero cost, and all these stored vectors will now be different. The process of discarding from further considerations those vectors $\{P_i\}$ for which $x_{i-n} \neq s'_{i-n}$, now sufficiently ensures that no two or more stored vectors $\{X_i\}$ can subsequently become the same. That is, the stored vectors $\{X_i\}$ are here prevented from merging.

In the case where no vectors of $\{X_i\}$ are available for selection during the detection process, a vector with arbitrary components $\{x_k\}$ is selected and its associated cost is set to a very high value so that this vector is discarded in the next detection process.

In the computer simulation tests to determine the tolerance to Gaussian noise of system 1, the linear prefilter shown in Fig. 5.2 is included as part of the channel of Fig. 2.3, so that the near-maximum likelihood detector now operates directly over the resultant 'channel' with sampled impulse response $y'_0, y'_1, \cdots y'_g$ (eqn. 5.5). The sampled impulse responses of the resultant 'channels' to be used here are those given in Tables 2.6 and 2.7. This arrangement is very convenient for the computer simulation tests, and it does not affect the results since the noise components at the prefilter output here have the same statistical properties as those at its input.

5.4 System 2

In system 1, the signal processor of Fig. 5.1 is implemented as a simple near-maximum likelihood detector with a linear prefilter inserted at its input. The improvement in tolerance to noise over the optimum nonlinear equalizer described in section 2.6.3 has been shown to be very satisfactory.(C42) However, as is mentioned in section 5.1, the weaknesses of using the linear prefilter used in system 1 are that, a long starting up procedure is required to set the tap-gain values of this prefilter to their correct values, and that a large number of taps are required to realise this prefilter so that a large number of multiplications are now required to evaluate a sample value at its output. The aim here is therefore to replace the linear prefilter used in system 1 by a simpler linear prefilter, or else to discard it altogether and use a more complicated near-maximum likelihood detector. System 2 is the basic system here in that, each of the following systems to be described in the following sections is a further development of system 2.

The signal processor of system 2 employs the arrangement where no prefilter is inserted ahead of the near-maximum likelihood detector. Before proceeding to describe the operation of this signal processor, consider first the detector used in system 1. It can be seen from section 5.3 that, the detector of system 1 is such that, only 16m possible combinations of the n+1 components $x_{i-n}, x_{i-n+1}, \cdots, x_i$ are available for the selection of the m possible vectors $\{X_i\}$ associated with the smallest costs. However, since each vector X_i here has n components $x_{i-n+1}, x_{i-n+2}, \cdots, x_i$, there are altogether 16^n ($\gg 16m$) possible vectors of $\{X_i\}$, bearing in mind that x_h ($h > 0$) has one of the 16 possible values of the data symbol s_h. Furthermore, the selection of the m vectors

$\{X_i\}$ here is carried out after the receipt of the sample r'_i (eqn. 5.7) which contains the components $s_i y'_0$, $s_{i-1} y'_1$, $\cdots$, $s_{i-n} y'_n$ of the corresponding signal elements carrying the respective data symbols s_i, s_{i-1}, $\cdots$, s_{i-n}. It therefore follows that, after the receipt of $s_i y'_0$, $s_{i-1} y'_1$, $\cdots$, $s_{i-n} y'_n$, some possible combinations of x_i, x_{i-1}, $\cdots$, x_{i-n} are discarded and are thus not available for the subsequent detection processes. Consequently, if the magnitudes of y'_0, y'_1, $\cdots$, y'_f $(f \geq 0)$ are very small, then some possible combinations of x_i, x_{i-1}, $\cdots$, x_{i-f+1} are here discarded before any components of significant magnitude dependent on s_i, s_{i-1}, $\cdots$, s_{i-f+1} are received, so that the m selected vectors $\{X_i\}$ can be very poorly chosen. Nevertheless, the linear prefilter used in system 1 is such that y'_0 is always one of the larger components in the sampled impulse response of the channel and prefilter, and so a satisfactory performance is always achieved here. In system 2, however, the detector is required to operate directly over the telephone channel whose sampled impulse response often has some small components at its front end, as can be seen from Tables 2.4 and 2.5. The detector of system 1 therefore may not operate satisfactorily if it is used in the arrangement of system 2. One approach to avoid the discard of some possible combinations of x_i, x_{i-1}, $\cdots$, x_{i-f+1} before any components of significant magnitude dependent on s_i, s_{i-1}, $\cdots$, s_{i-f+1} are received, is to include all the possible combinations of x_i, x_{i-1}, $\cdots$, x_{i-f+1} in the m selected vectors $\{X_i\}$. The drawback of this arrangement is that there is now a very considerable increase in both the number of stored vectors and in the number of operations per received signal element. Another approach is that employed in the detector of system 2 now to be described.

Basically, the detector of system 2 differs from that of system 1 in that the selection of the m possible vectors $\{X_i\}$ associated with the smallest costs is here carried out after the first significant component $s_i y_f$ of the signal element carrying the data symbol s_i has been received. The signal processor of system 2 now has the arrangement shown in Fig. 5.3 where the data symbol s_{i-n} is detected after the receipt of the sample value r_{i+f} (eqn. 5.1). The signal processor here operates as follows. After the receipt of the sample value r_{i+f}, the intersymbol interference canceller (Fig. 5.3) removes from the samples $r_i, r_{i+1}, \cdots, r_{i+f}$, detected values of all components involving data symbols whose final detected values $s'_{i-n-1}, s'_{i-n-2}, \cdots, s'_{i-g}$ have already been determined. That is, the intersymbol interference canceller operates on $r_i, r_{i+1}, \cdots, r_{i+f}$ to give at the detector input the following complex-valued quantities

$$r'_{i,0} = r_i - \sum_{h=n+1}^{g} s'_{i-h} y_h$$

$$r'_{i,1} = r_{i+1} - \sum_{h=n+2}^{g} s'_{i+1-h} y'_h$$

$$\vdots$$

$$r'_{i,f} = r_{i+f} - \sum_{h=n+f+1}^{g} s'_{i+f-h} y_h \qquad (5.14)$$

where $y_0, y_1, \cdots, y_g$ are the g+1 components of the telephone channel sampled impulse response, $\{r_k\}$ are as defined by eqn. 5.1, and y_f is the first significant component of the channel sampled impulse response. The selection of the component y_f will be further discussed. Assuming for the moment that $s'_{i-h} = s_{i-h}$ for $h = n+1, n+2, \cdots, g$, eqn. 5.14

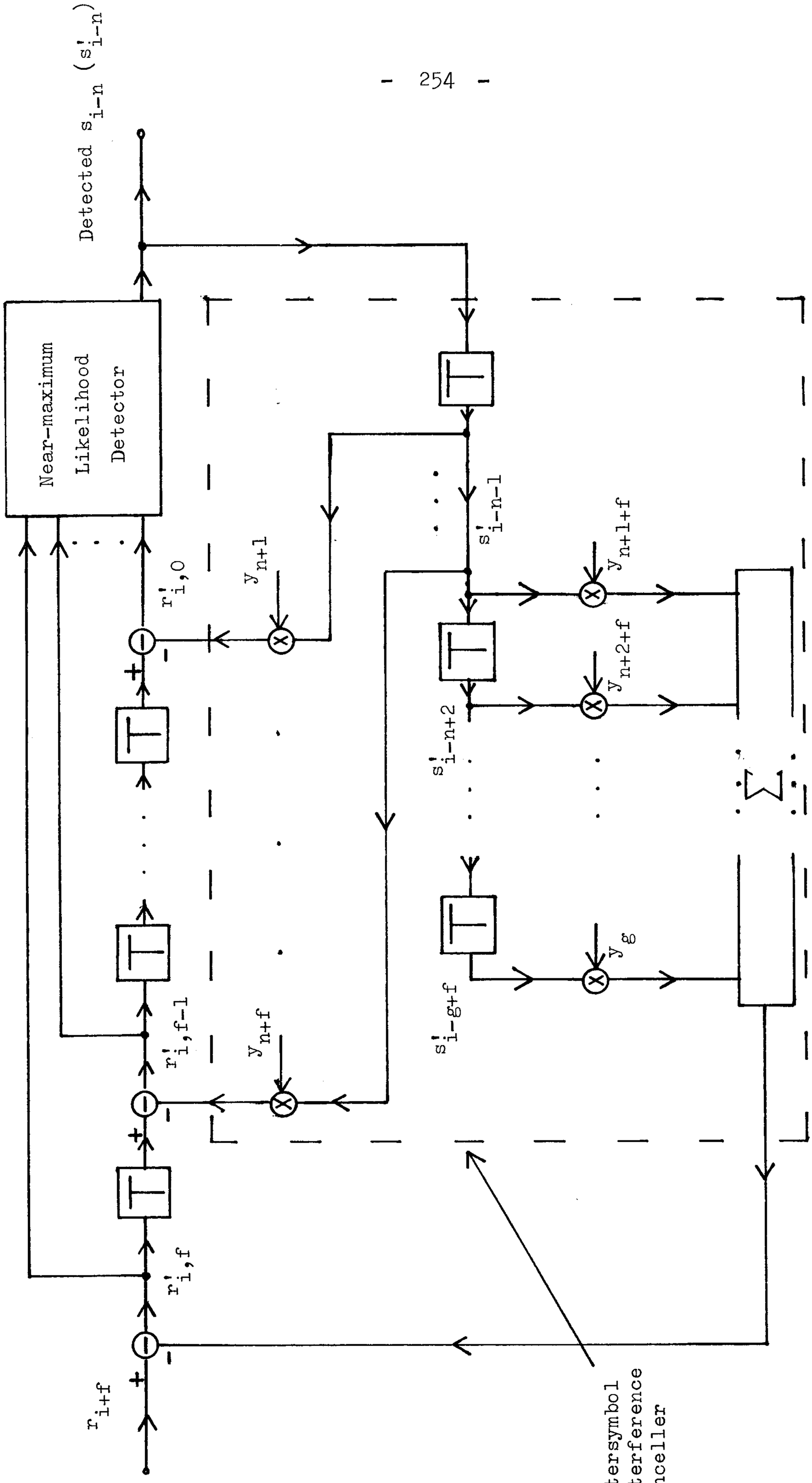


Fig. 5.3 Signal processor of system 2

now reduces to

$$r'_{i,0} = \sum_{h=0}^{n} s_{i-h} y_h + w_i$$

$$r'_{i,1} = \sum_{h=0}^{n+1} s_{i+1-h} y_h + w_{i+1}$$

$$\vdots$$

$$r'_{i,f} = \sum_{h=0}^{n+f} s_{i+f-h} y_h + w_{i+f} \qquad (5.15)$$

where $\{w_k\}$ are the noise components whose real and imaginary parts are statistically independent Gaussian random variables with zero mean and variance σ^2.

The detector (Fig. 5.3) of system 2 operates by ignoring the components associated with $s_{i+1}, s_{i+2}, \cdots, s_{i+f}$ in the sample values $r'_{i,0}, r'_{i,1}, \cdots, r'_{i,f}$, and this is now described as follows. Just prior to the receipt of the sample r_{i+f} (eqn. 5.1), the detector here holds in store m n-component vectors $\{X_{i-1}\}$ (eqn. 5.8) together with the associated costs $\{C_{i-1}\}$. The cost C_{i-1} here is as defined by eqn. 5.10 to be $c_1 + c_2 + \cdots + c_{i-1}$ but the quantity c_k used in system 2 is given by

$$c_k = \left| r'_{k,0} - \sum_{h=0}^{n} x_{k-h} y_h \right|^2 \qquad (5.16)$$

for any possible value of k, where $r'_{k,0}$ is as defined by eqn. 5.14, and x_j has one of the 16 possible values of the data symbol s_j. On receiving r_{i+f}, the detector expands each vector X_{i-1} into the corresponding 16 vectors $\{P_i\}$ (eqn. 5.9) having the 16 possible values of x_i, and

it evaluates the cost C_i for each of the 16 vectors $\{P_i\}$. The cost C_i here is evaluated by using eqns. 5.11 and 5.16. Furthermore, the detector also evaluates the quantity C_i' for each of the 16m expanded vectors $\{P_i\}$, where

$$\begin{aligned} C_i' &= C_i && \text{, if } f = 0 \\ &= C_i + \sum_{h=1}^{f} \left| r'_{i,h} - \sum_{j=0}^{n+h} x_{i+h-j} y_j \right|^2 && \text{, if } f > 0 \end{aligned} \qquad (5.17)$$

$x_k = 0$ for $k \leq 0$, $k > i$, and the $\{r'_{i,h}\}$ are as defined by eqn. 5.14. Thus, in the computation of the quantity C_i', all terms $\{x_{i+h-j} y_j\}$ for $h-j = 1, 2, \cdots, f$ are neglected here, so that as far as the detector is concerned, the kth signal element has not arrived until its (f+1)th component $s_k y_f$ is received. Having evaluated the value of C_i' for each of the 16m vectors $\{P_i\}$, the detected data symbol s'_{i-n} is then taken as the value of x_{i-n} in the vector P_i having the smallest value of C_i'. All vectors $\{P_i\}$ for which $x_{i-n} \neq s'_{i-n}$ are now discarded, and the first component of each of the remaining vectors $\{P_i\}$ is omitted to give the corresponding n-component vectors $\{X_i\}$. Each vector X_i is, of course, associated with the same C_i and C_i' as those of the corresponding P_i. The detector now selects the m vectors $\{X_i\}$ associated with the smallest values of $\{C_i'\}$, subject to the constraint that only the 16 vectors $\{X_i\}$ originating from the previous best vector X_{i-1} are available for the selection of the second vector X_i here, where the best vector X_{i-1} here is referred to the vector having the smallest value of C'_{i-1}. These m selected vectors $\{X_i\}$, together with their associated costs $\{C_i\}$ are then stored, and the detector is now ready for the next detection process.

The starting procedure here is the same as that of system 1. That is, all the n components of each of the m initial stored vectors $\{X_n\}$ are set to the values of the corresponding data symbols, and the cost is set to zero for the first vector and to a very high value for each of the remaining m-1 vectors. When no vectors of $\{X_i\}$ are available for selection during the detection process, an arbitrary vector X_i is selected and its cost is set to a very high value so that this vector is discarded in the next detection process.

Thus, the detector of system 2 differs from that of system 1 in that the quantity C_i' (eqn. 5.17) is used in place of the cost C_i to select the m vectors $\{X_i\}$. In the special case when f = 0 (that is, when y_0 is the first significant component of the channel sampled impulse response), C_i' becomes identical to C_i and the detector here reduces exactly to that of system 1. Furthermore, if f > 0, then the values of $\{r'_{i,h}\}$ given by eqn. 5.15 may be substituted into eqn. 5.17 so that the value of C_i' now becomes

$$C_i' = C_i + \sum_{h=1}^{f} \left| \sum_{j=0}^{n+h} (s_{i+h-j} - x_{i+h-j}) y_j + w_{i+h} \right|^2$$

$$= C_i + \sum_{h=1}^{f} \left| \sum_{j=h}^{n+h} (s_{i+h-j} - x_{i+h-j}) y_j + \sum_{j=0}^{h-1} s_{i+h-j} y_j + w_{i+h} \right|^2$$

$$= C_i + \sum_{h=1}^{f} \left| \sum_{j=0}^{n} (s_{i-j} - x_{i-j}) y_{h+j} + I_h + w_{i+h} \right|^2 \qquad (5.18)$$

and $$I_h = \sum_{j=0}^{h-1} s_{i+h-j} y_j \qquad (5.19)$$

for h = 1, 2, ·· , f where the $\{I_h\}$ are the intersymbol interference components caused by neglecting the presence of $s_{i+1}, s_{i+2}, \cdots, s_{i+f}$ in $r'_{i,1}, r'_{i,2}, \cdots, r'_{i,f}$ of eqn. 5.15. It can now be seen from

eqn. 5.18 that, by using a larger value of f, more components of the received signal elements associated with $s_{i-n+1}, s_{i-n+2}, \cdots, s_i$ are included in the value of C_i', so that more information about the data symbols $s_{i-n+1}, s_{i-n+2}, \cdots, s_i$ is now available for the selection of the m vectors $\{X_i\}$. However, it can also be seen from eqns. 5.18 and 5.19 that, by using a larger value of f, more intersymbol interference components $\{I_h\}$ are included in C_i' so that the effective noise level is now increased, bearing in mind that the effective noise components may be taken as $I_h + w_{i+h}$ in eqn. 5.18. Consequently, an appropriate value of f must now be used in the detection process so as to achieve the best performance achievable by system 2. A crude analysis of the value of f to be used at high signal to noise ratios is given as follows. At high signal to noise ratios when the error rates are low, most of the data symbols $\{s_i\}$ are detected correctly. It is now very likely that the vector X_i whose components are all correct (that is, $x_h = s_h$ for $h = i-n+1, i-n+2, \cdots, i$) is available for the selection process that selects the m vectors $\{X_i\}$ associated with the smallest $\{C_i'\}$. It appears intuitively that, the probability of error in detection is now likely to be determined to a large extent by the probability of discarding the vector X_i whose components are all correct during the selection process for the m vectors $\{X_i\}$. Consider now the 16 vectors $\{X_i\}$ whose first n-1 components are all correct. The value of C_i' (eqn. 5.18) for each of these 16 vectors now reduces to

$$C_i' = C_i + \sum_{h=1}^{f} \left| (s_i - x_i) y_h + I_h + w_{i+h} \right|^2 \qquad (5.20)$$

where x_i may have any of the 16 possible values of s_i, and I_h is as given by eqn. 5.19. As eqn. 5.20 suggests, the inclusion of a component $s_i y_h$ is always accompanied by the inclusion of an intersymbol interference

component I_h in the value of C_i' which now determines which of the 16 vectors $\{X_i\}$ having the 16 possible values of x_i are to be discarded during the selection process for $\{X_i\}$. Clearly, the inclusion of $s_i y_h$ in C_i' allows the detector to know more about the value of s_i whereas the inclusion of I_h in C_i' may be regarded as the additional 'noise' added to the value of $s_i y_h$. It therefore follows that if the magnitude of $s_i y_h$ is larger than that of I_h, for $h = 1, 2, \cdots, f$, then the value of C_i' is likely to be in favour of the system performance. Consequently, it appears that the value of f used here should be selected to be such that

$$|s_i y_f| > |I_f| = \left|\sum_{k=0}^{f-1} s_{i+f-k} y_k\right| \qquad (5.21)$$

and the larger the difference between $|s_i y_f|$ and $|I_f|$, the higher may be the tolerance to noise of the detector. Furthermore, since

$$\sum_{k=0}^{f-1} |s_{i+f-k} y_k| \geq \left|\sum_{k=0}^{f-1} s_{i+f-k} y_k\right| \qquad (5.22)$$

the inequality of (5.21) is always satisfied if

$$|s_i y_f| > \sum_{k=0}^{f-1} |s_{i+f-k} y_k| \qquad (5.23)$$

or

$$|s_i||y_f| > \sum_{k=0}^{f-1} |s_{i+f-k}||y_k| \qquad (5.24)$$

Since the smallest possible value of $|s_i|$ is $|\pm 1 \pm j|$ or $\sqrt{2}$, and the largest possible value of $|s_{i+f-k}|$ is $|\pm 3 \pm 3j|$ or $3\sqrt{2}$, therefore if the quantity

$$d = |y_f| - 3 \sum_{k=0}^{f-1} |y_k| \qquad (5.25)$$

is larger than zero, then the inequalities of (5.24) and hence (5.21)

are always satisfied. Thus, the value of f and hence the first significant component y_f to be used in the detection process here, at high signal to noise ratios, should be such that the quantity d defined by eqn. 5.25 is maximized. It is emphasized that, the above derivation is not rigorous and that the mechanism of the detection process here is more complicated than the simplified version described above.

Computer simulation tests have been carried out to determine the tolerance to additive white Gaussian noise of system 2 operating over channel A (Table 2.4), and the results are shown in Fig. 5.4. The 95% confidence limits of these results are about ± 0.5 dB, and the definitions of the bit error rate and the signal to noise ratio here are as defined by eqns. 5.3 and 5.4 respectively. The performances of system 1 and the optimum nonlinear equalizer (section 2.6) are also shown here. Table 5.1 gives the calculated values of d (eqn. 5.25) for $h = 0$, 1, and 2 for channel A.

f	$\lvert y_f \rvert$	$3 \sum_{k=0}^{f-1} \lvert y_k \rvert$	d (eqn. 5.25)
0	0.025	0.000	0.025
1	0.186	0.075	0.111
2	0.492	0.633	-0.141

Table 5.1 Calculated values of d (eqn. 5.25) for system 2 operating over channel A (Table 2.4).

As Fig. 5.4 shows, system 2 with f = 1 appears to have a better tolerance to noise than that of the system with f = 2 at high signal to noise ratios where the error rates drop below 10^{-3}. At low signal to noise ratios, the situation is reversed and system 2 with f = 2 now appears to have a better tolerance to noise than that of the system with f = 1. Furthermore, it can be seen from Table 5.1 that channel A with f = 1 has a larger value of d but a smaller value of $|y_f|$ as compared to those of the same channel with f = 2. It therefore appears, for channel A, that with a larger value of $|y_f|$, system 2 has a higher tolerance to noise at low signal to noise ratios, and with a larger value of d, system 2 has a higher tolerance to noise at high signal to noise ratios. The exact explanation to these behaviours can be very complex but a crude explanation may be given as follows. It has been discussed before that the inclusion of $s_i y_f$ in C_i' (eqn. 5.18) is in favour of the system performance, but this is always accompanied by the inclusion of the intersymbol interference component I_f (eqn. 5.19) and the noise component w_{i+f} which both degrade the system performance. At low signal to noise ratios, the noise components $\{w_i\}$ are very often so much larger than the intersymbol interference components $\{I_f\}$ that the effect of $\{I_f\}$ on the system performance becomes relatively insignificant Consequently, the system performance is now determined to a large extent by the magnitude of the component $s_i y_f$ and so system 2 with a larger value of $|y_f|$ is now likely to have a better performance. At high signal to noise ratios when the noise components $\{w_i\}$ are often negligibly small, the intersymbol interference components $\{I_f\}$ now have a significant effect on the system performance. Since I_f may be regarded as a noise component added to $s_i y_f$, therefore system 2 with a larger value of $|s_i y_f| - |I_f|$ or d (eqn. 5.25) is now likely to have a better performance as has been explained before.

Nevertheless, it can be seen from Fig. 5.4 that the tolerance to noise of system 2 operating over channel A is not only poorer than that of system 1, but is also inferior to that of the optimum nonlinear equalizer. Modifications to this system are therefore needed. One main reason for having a poor performance here appears to be that the value of $|y_f|$ is not very much larger than that of $3\sum_{k=0}^{f-1}|y_k|$ for channel A (as can be seen from Table 5.1), so that the advantage of including the component $s_i y_f$ in C_i' (eqn. 5.18) is now greatly reduced due to the large intersymbol interference component I_f (eqn. 5.19) incurred here. Two approaches are considered to overcome this weakness. In the first approach, a simple linear prefilter is inserted ahead of the detector to, hopefully, increase the value of d (eqn. 5.25), and this leads to the developments of systems 3, 4, and 5. The second approach uses no prefilter but a modified near-maximum likelihood detection process, and this leads to the developments of systems 6, 7, and 8.

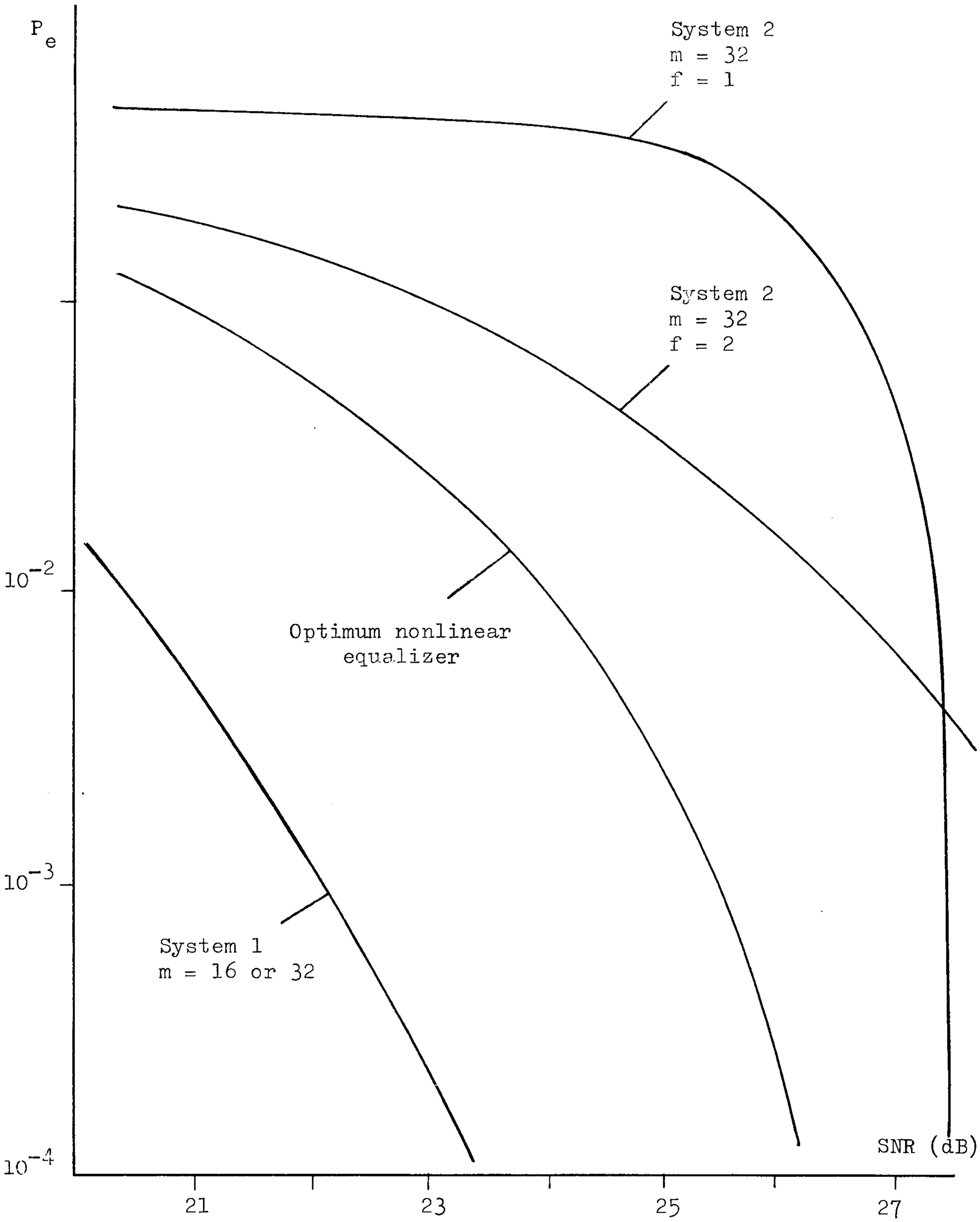


Fig. 5.4 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 2 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16.

5.5 System 3

It has been suggested in section 5.4 that, the detector of system 2 may have a higher tolerance to noise at high signal to noise ratios if the channel has a larger value of $|y_f| - 3 \sum_{k=0}^{f-1} |y_k|$, where $y_0, y_1, \cdots, y_g$ are the channel sampled impulse response and y_f is the first significant component here. System 3 is a modification of system 2 in that it uses a simple linear prefilter to maximize the first significant component of the sampled impulse response of the channel and prefilter. Fig. 5.5 shows the model of the signal processor of system 3.

In Fig. 5.5, the linear prefilter is implemented as a linear feedforward transversal filter with p+1 taps. let the z-transform of this linear prefilter be

$$F(z) = f_0 + f_1 z^{-1} + \cdots + f_p z^{-p} \qquad (5.26)$$

where $\{f_k\}$ are the tap gains of the prefilter. In system 3, the linear prefilter is selected to be matched to the first p+1 components of the channel sampled impulse response. That is, the tap-gain values of the prefilter here are set to

$$f_k = y^*_{p-k} \qquad (5.27)$$

for $k = 0, 1, \cdots, p$ where y^*_j is the complex conjugate of y_j.

The advantage of this partially-matched filter is its simplicity in terms of practical implementation. There are here only p+1 taps required where p is usually a very small number. Furthermore, as eqn. 5.27 suggests, these p+1 tap gains can very easily be obtained from the channel sampled impulse response which in turn can be estimated relatively simply and quite accurately.(A9;C39,C43)

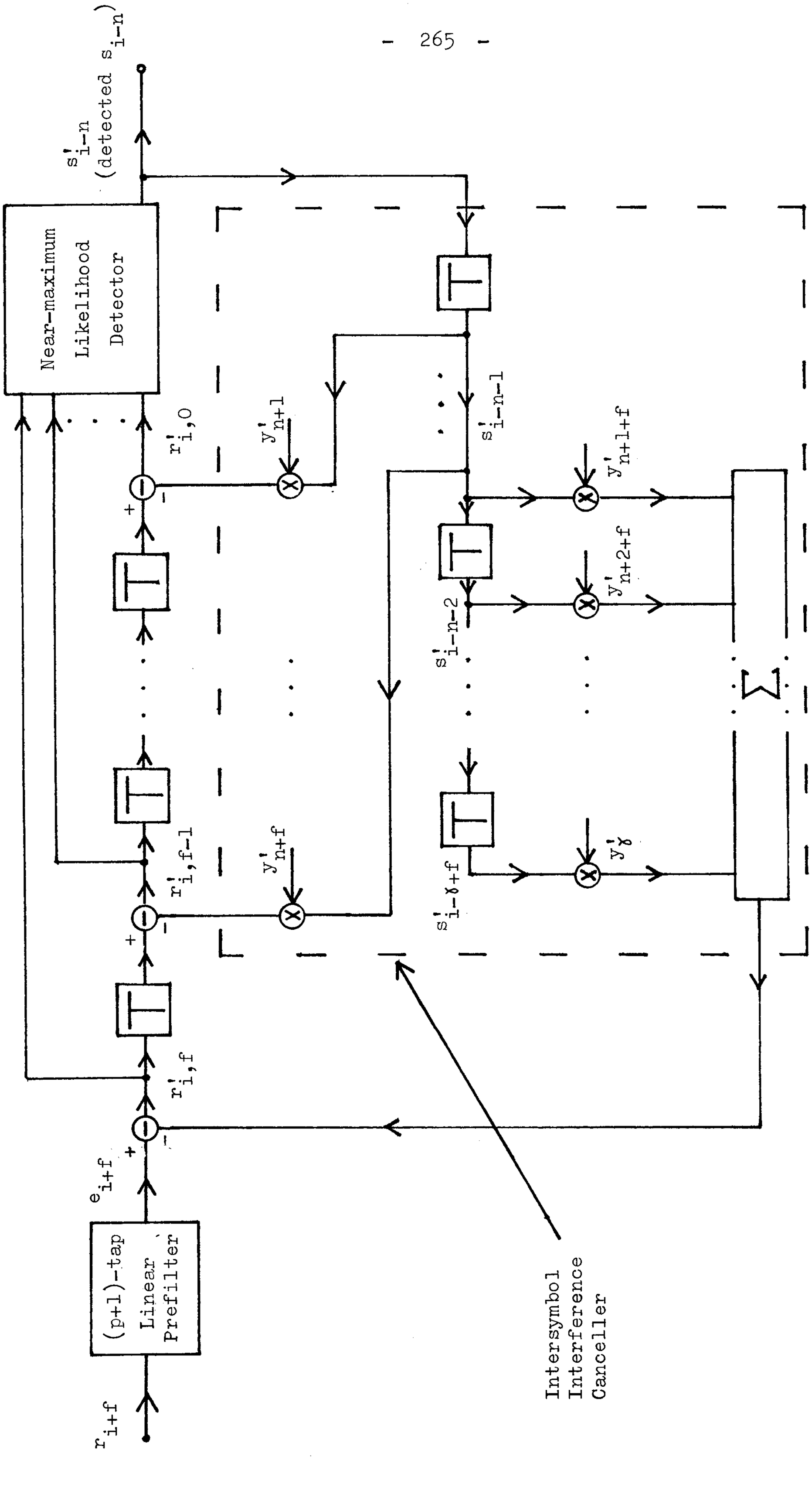


Fig. 5.5 Signal processor of any of the systems 3, 4, and 5.

The sample value e_i at the output of the prefilter shown in Fig. 5.5, at time $t = iT$, is

$$e_i = \sum_{h=0}^{p} r_{i-h} f_h \tag{5.28}$$

where $\{r_k\}$ are the sample values at the prefilter input and $\{f_k\}$ are the tap gains of the prefilter. Substituting the values of $\{r_k\}$ from eqn. 5.1 into eqn. 5.28, the value of e_i can be rearranged to become

$$e_i = \sum_{h=0}^{\gamma} s_{i-h} y'_h + w'_i \tag{5.29}$$

where $$\gamma = g + p + 1 \tag{5.30}$$

$$w'_i = \sum_{k=0}^{p} w_{i-k} f_k \tag{5.31}$$

$$y'_h = \sum_{k=0}^{p} y_{h-k} f_k \tag{5.32}$$

for $h = 0, 1, \cdots, \gamma$ and $y_j = 0$ for $j < 0$, $j > g$. In eqn. 5.29, $y'_0, y'_1, \cdots, y'_\gamma$ are the $\gamma+1$ components of the channel and prefilter sampled impulse response, and $\{w'_i\}$ are the noise components at the output of the prefilter. Thus, as far as the detector and the intersymbol interference canceller (Fig. 5.5) are concerned, the ith transmitted signal element is now time-dispersed into $\gamma+1$ components $s_i y'_0, s_i y'_1, \cdots, s_i y'_\gamma$. Furthermore, since the real and imaginary parts of the noise components $\{w_i\}$ at the prefilter input are Gaussian random variables with zero mean and variance σ^2, it can be shown[A9] that, the real and imaginary parts of the noise components $\{w'_i\}$ at the prefilter output are also Gaussian random variables with zero mean and variance

$$\eta^2 = \sigma^2 k_p^2 \tag{5.33}$$

where

$$k_p^2 = \sum_{k=0}^{p} |f_k|^2 \qquad (5.34)$$

and $\{f_k\}$ are the tap gains of the prefilter. Since the linear prefilter of system 3 is matched to the first p+1 components of the channel sampled impulse response, it necessarily maximizes the energy of the component $s_{i-p}y'_p$ in e_i (eqn. 5.29) for a given noise level in e_i.[A9] That is, for a given value of the noise variance σ^2 (eqn. 5.33), the value of $|y'_p|$ is maximized in system 3.

Having described the properties of the linear prefilter used in system 3, the operations of the near-maximum likelihood detector and the intersymbol interference canceller of system 3 are now described as follows. Basically, they operate in a similar way as those of system 2. Thus, after receiving the sample value e_{i+f} at the output of the prefilter shown in Fig. 5.5, the intersymbol interference canceller operates on the samples $e_i, e_{i+1}, \cdots, e_{i+f}$ to give the complex-valued quantities

$$r'_{i,0} = e_i - \sum_{h=n+1}^{\gamma} s'_{i-h}y'_h$$

$$r'_{i,1} = e_{i+1} - \sum_{h=n+2}^{\gamma} s'_{i+1-h}y'_h$$

$$\vdots$$

$$r'_{i,f} = e_{i+f} - \sum_{h=n+f+1}^{\gamma} s'_{i+f-h}y'_h \qquad (5.35)$$

where $s'_{i-n-1}, s'_{i-n-2}, \cdots, s'_{i-\gamma}$ are the previously detected data symbols, $e_i, e_{i+1}, \cdots, e_{i+f}$ are as defined by eqn. 5.29, and y'_f

is the first significant component of the sampled impulse response of the channel and prefilter. When $s'_{i-h} = s_{i-h}$ for $h = n+1, n+2, \cdots, \gamma$ eqn. 5.35 reduces to

$$r'_{i,0} = \sum_{h=0}^{n} s_{i-h} y'_h + w'_i$$

$$r'_{i,1} = \sum_{h=0}^{n+1} s_{i+1-h} y'_h + w'_{i+1}$$

$$\vdots$$

$$r'_{i,f} = \sum_{h=0}^{n+f} s_{i+f-h} y'_h + w'_{i+f} \qquad (5.36)$$

where the noise components $\{w'_k\}$ are as defined by eqn. 5.31.

The detector (Fig. 5.5) operates on its input samples $r'_{i,0}$, $r'_{i,1}, \cdots, r'_{i,f}$ to give at its output the detected data symbol s'_{i-n} so that there is now a delay in detection of n+f sampling intervals. Thus, just prior to the receipt of $r'_{i,f}$, the detector holds in store m n-component vectors $\{X_{i-1}\}$ (eqn. 5.8) together with the associated costs $\{C_{i-1}\}$. The cost C_{i-1} here is as defined by eqn. 5.10 to be $c_1 + c_2 + \cdots + c_{i-1}$, but the quantity c_k is now given by

$$c_k = \left| r'_{k,0} - \sum_{h=0}^{n} x_{k-h} y'_h \right|^2 \qquad (5.37)$$

for any possible value of k, where x_j has one of the 16 possible values of the data symbol s_j. On receiving $r'_{i,f}$, each vector X_{i-1} is expanded into 16 (n+1)-component vectors $\{P_i\}$ (eqn. 5.9) having the 16 possible values of x_i. The detector then evaluates the cost C_i for each of the 16m expanded vectors $\{P_i\}$ by using eqns. 5.11 and 5.37. It then

proceeds to evaluate the quantity C_i' for each of these vectors, where

$$C_i' = C_i \quad , \quad \text{if} \quad f = 0$$

$$= C_i + \sum_{h=1}^{f} \left| r_{i,h}' - \sum_{j=0}^{n+h} x_{i+h-j} y_j' \right|^2 \quad , \quad \text{if} \quad f > 0 \qquad (5.38)$$

$x_k = 0$ for $k \leq 0$, $k > i$, and the $\{r_{i,h}'\}$ and $\{y_j'\}$ are as defined by eqns. 5.35 and 5.32 respectively. The detector then selects the vector P_i associated with the smallest value of C_i', and takes the detected data symbol s_{i-n}' to have the value of x_{i-n} in this vector. All vectors $\{P_i\}$ for which $x_{i-n} \neq s_{i-n}'$ are now discarded and the first component of each of the remaining $\{P_i\}$ is omitted to give the corresponding n-component vectors $\{X_i\}$. The detector then proceeds with the selection of the m vectors $\{X_i\}$ associated with the smallest $\{C_i'\}$ with the constraint that, only those vectors originating from the previous best vector X_{i-1} (having the smallest C_{i-1}') are available for the selection of the second vector X_i here. These m selected vectors $\{X_i\}$ together with their associated costs $\{C_i\}$ are then stored and the detector is now ready for the next detection process.

Thus, while the detector and intersymbol interference canceller of system 2 operate over the channel sampled impulse response with components $\{y_j\}$, the detector and intersymbol interference canceller of system 3 operate over the channel and prefilter sampled impulse response with components $\{y_j'\}$. The detection process of system 3 is otherwise exactly the same as that of system 2. Furthermore, it can be seen from eqns. 5.31, 5.33, and 5.34 that, if the tap gains $\{f_k\}$ of the prefilter of system 3 are scaled to become $\{f_k/k_p\}$, where k_p is as defined by eqn. 5.34, then the noise variance remains unchanged at the output of the prefilter here. The sampled impulse response of

the channel and prefilter now has the components $\{y'_j/k_p\}$ as can be seen from eqn. 5.32. Consequently, the arguments used to select the first significant component y_f in system 2 may be carried over here to select the first significant component y'_f/k_p (and hence y'_f), at high signal to noise ratios. That is, at high signal to noise ratios, the first significant component y'_f to be used in the detection process of system 3 should be selected to maximize the quantity

$$d = \left|\frac{y'_f}{k_p}\right| - 3\sum_{k=0}^{f-1}\left|\frac{y'_k}{k_p}\right| \qquad (5.39)$$

which is obtained from eqn. 5.25 (for system 2) by replacing y_j by y'_j/k_p for $j = 0, 1, \cdots, f$, and the constant k_p is as defined by eqn. 5.34. The values of d given by eqn. 5.39 have been calculated for system 3 operating over channel A (Table 2.4) at various values of p and f, where p+1 is the number of taps used in the prefilter of system 3 shown in Fig. 5.5, and the results are shown in Table 5.2. It can be seen from Tables 5.1 and 5.2 that, the largest value of d for system 3 is about the same as that for system 2. Consequently, at high signal to noise ratios, the best tolerance to noise achievable by system 3 is likely to be about the same as that achievable by system 2, when operating over channel A (Table 2.4).

Nevertheless, computer simulation tests are carried out to determine the tolerance to additive white Gaussian noise of system 3 operating over channel A (Table 2.4), and the results are shown in Fig. 5.6. The 95% confidence limits of these results are about ± 0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 5.3 and 5.4 respectively.

p	f	$\left\lvert \frac{y'_f}{k_p} \right\rvert$	$3 \sum_{k=0}^{f-1} \left\lvert \frac{y'_k}{k_p} \right\rvert$	d (eqn. 5.39)
1	0	0.025	0.000	0.025
	1	0.188	0.075	0.113
	2	0.512	0.639	-0.127
2	0	0.023	0.000	0.023
	1	0.183	0.069	0.114
	2	0.527	0.618	-0.091
3	0	0.018	0.000	0.018
	1	0.146	0.054	0.092
	2	0.472	0.492	-0.020
4	0	0.012	0.000	0.012
	1	0.087	0.036	0.051
	2	0.274	0.297	-0.023

Table 5.2 Calculated values of d (eqn. 5.39) for system 3 operating over channel A (Table 2.4). p+1 is the number of taps of the prefilter shown in Fig. 5.5, and k_p is the constant defined by eqn. 5.34.

It can be seen from Table 5.2 and Fig. 5.6 that, system 3 with a larger value of $\left| y'_f/k_p \right|$ appears to have a higher tolerance to noise at low signal to noise ratios where the error rates are higher than 10^{-3}. As the error rates drop to about 10^{-3}, system 3 with the various arrangements shown in Fig. 5.6 (except those with p = 4) appear to have roughly the same tolerance to noise. At lower error rates, Table 5.2

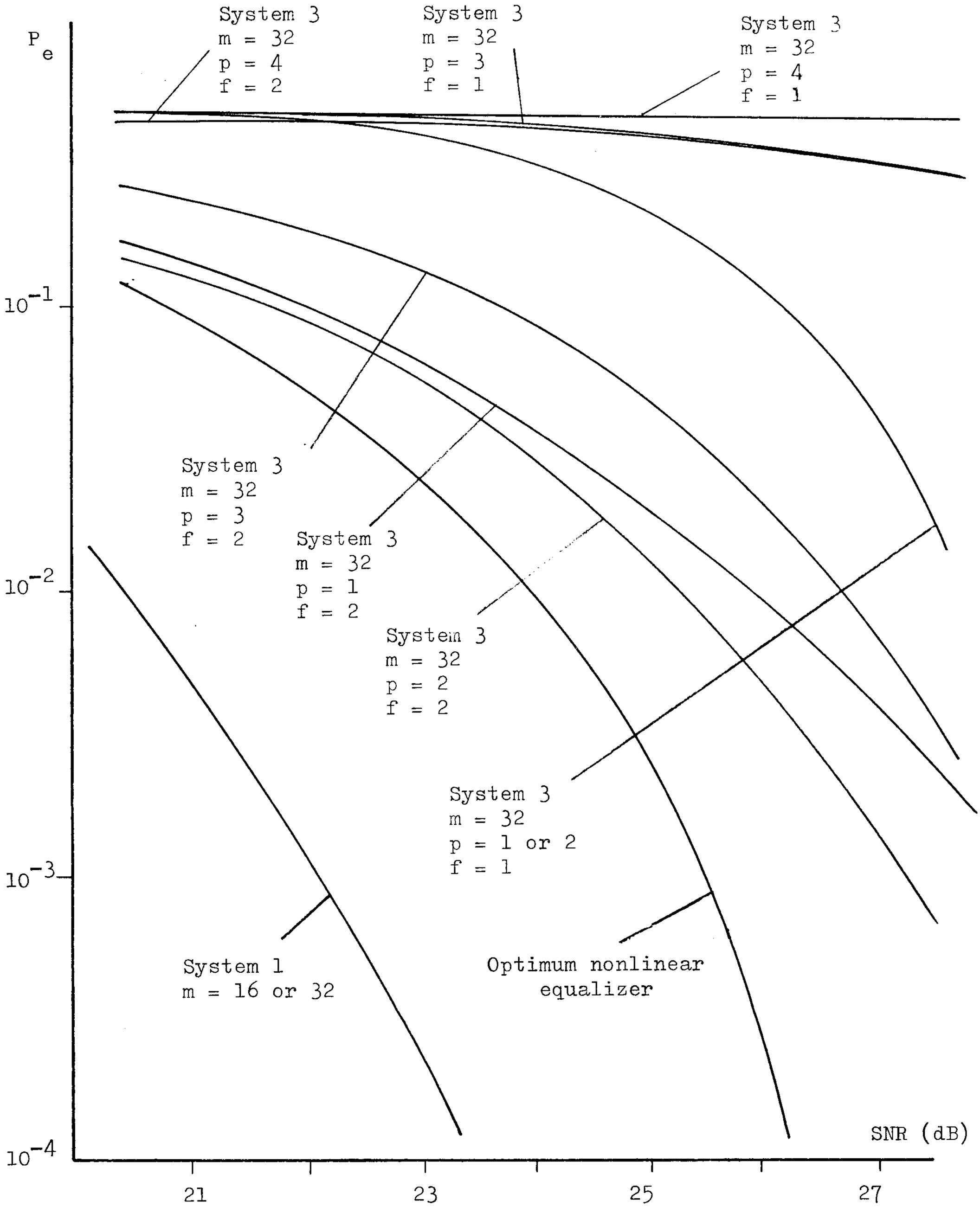


Fig. 5.6 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 3 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16.

and Fig. 5.6 show that system 3 with a larger value of d (eqn. 5.39) tends to have a higher tolerance to noise here. System 3 with p = 4, f = 1 and p = 4, f = 2 appears to have relatively small values of $|y'_f/k_p|$ and d, and relatively poor tolerances to noise over channel A here. These results are in agreement with the crude explanation given to the computer simulation results of system 2 operating over channel A in section 5.4, bearing in mind that the corresponding values of $|y'_f/k_p|$ and d (eqn. 5.39) for system 2 are y_f and d (eqn. 5.25) respectively. It can be seen from Tables 5.1, 5.2 and Figs. 5.4, 5.6 that system 3 can have a slightly larger value of $|y'_f/k_p|$ and a slightly better tolerance to noise at low signal to noise ratios in relation to those of system 2. Nevertheless, the largest value of d for system 3 here appears to be about the same as that for system 2, and the best tolerance to noise achievable at high signal to noise ratios for system 3 also appears to be about the same as that for system 2. The weakness of system 3 therefore appears to be that, by increasing the magnitude of the first significant component of the channel sampled impulse response, the prefilter here also increases the magnitudes of the preceeding components, so that the value of d (eqns. 5.25 and 5.39) is now changed only by an insignificant amount, bearing in mind that the value of d here is basically a measure of the relative magnitude of the first significant component and the preceeding components. Consequently, the improvement in the tolerance to additive white Gaussian noise gained by having the prefilter here (being a partially-matched filter) is limited to be at low signal to noise ratios and is quite small for channel A here. Further modifications to system 3 are therefore needed.

5.6 System 4

Very often, the magnitudes of the first few components of the channel sampled impulse response are found to be such that

$$|y_0| < |y_1| < \cdots < |y_c| \tag{5.40}$$

where c is a small positive integer. Let the z-transform of these c+1 components be

$$V_1(z) = y_0 + y_1 z^{-1} + \cdots + y_c z^{-c} \tag{5.41}$$

The inequalities of (5.40) imply that all zeros (roots) of the z-transform $V_1(z)$ here must necessarily lie outside the unit circle in the z-plane. Consequently, this z-transform can easily be equalized by using a maximum-delay linear feedforward transversal equalizer[A9] which forces to zero all components preceeding the last of the sequence $y_0, y_1, \cdots, y_c$. This is the basis of system 4.

Thus, system 4 uses the same basic model of signal processor as that shown in Fig. 5.5 where a linear transversal filter is inserted ahead of a near-maximum likelihood detector. The linear prefilter used in system 4 is selected to perform approximately the function of a maximum-delay linear feedforward transversal equalizer mentioned above. That is, the z-transform of the (p+1)-tap linear prefilter (Fig. 5.5) here is given by

$$\begin{aligned} F(z) &= f_0 + f_1 z^{-1} + \cdots + f_p z^{-p} \\ &\simeq z^{-(c+p)} V_1^{-1}(z) \end{aligned} \tag{5.42}$$

where c is the positive integer defined in eqn. 5.40, and $z^{-(c+p)}$ is

the delay of the maximum-delay filter here.[A9] Thus, while the prefilter of system 3 is matched to the first p+1 components of the channel sampled impulse response, the prefilter of system 4 is approximately an inverse filter for the first c+1 components of the channel sampled impulse response. This further implies that, while the prefilter of system 3 maximizes the value of $|y'_p|$, the prefilter here (eqn. 5.42) may minimize the value of $3\sum_{k=0}^{c+p-1}|y'_k|$, where $y'_0, y'_1, \cdots, y'_\gamma$ are the sampled impulse response of the channel and prefilter given by eqn. 5.32 and γ is as given by eqn. 5.30.

The evaluation of the partially-inverse prefilter $F(z)$ (eqn. 5.42) of system 4 is now described as follows. Clearly, the z-transfrom $z^{-(c+p)}\ V_1^{-1}(z)$ can be more accurately represented by the prefilter $F(z)$ here if a larger value of p is used in this prefilter. However, it is also desirable to use a smaller value of p in $F(z)$ so that the components $y'_0, y'_1, \cdots, y'_{c+p}$ are affected to a smaller extent by the presence of the components $y_{c+1}, y_{c+2}, \cdots, y_{c+p}$, bearing in mind that the prefilter $F(z)$ here is a maximum-delay filter only for the components $y_0, y_1, \cdots, y_c$. It therefore follows that, a moderate value of p must be used here. Thus, in system 4, the value of p is selected to be the same as that of c, and eqn. 5.42 now becomes

$$\begin{aligned} F(z) &= f_0 + f_1 z^{-1} + \cdots + f_c z^{-c} \\ &\simeq z^{-2c}\ V_1^{-1}(z) \end{aligned} \tag{5.43}$$

The evaluation of $V_1^{-1}(z)$ in eqn. 5.43 is now described as follows. It can be seen from eqns. 5.40 and 5.41 that, a direct inversion of $V_1(z)$ using the long division operation will obviously lead to a divergent series which can not be realised by the (c+1)-tap prefilter $F(z)$ here. It is therefore necessary now to perform the inversion of $V_1(z)$ in

two stages.[A9] Thus, the z-transform $V_1(z)$ in eqn. 5.41 is first rearranged to become

$$V_1(z) = z^{-c} (y_c + y_{c-1}z + \cdots + y_0 z^c) \qquad (5.44)$$

so that its inverse now becomes

$$V_1^{-1}(z) = \frac{z^c}{(y_c + y_{c-1}z + \cdots + y_0 z^c)} \qquad (5.45)$$

The long division operation[A9] can now be used to evaluate the value of $V_1^{-1}(z)$ given by eqn. 5.45 to give

$$V_1^{-1}(z) = z^c (f'_0 + f'_1 z + \cdots + f'_c z^c + \cdots) \qquad (5.46)$$

where the values of f'_0, f'_1, $\cdots$ here form a convergent series, so that

$$V_1^{-1}(z) \simeq z^c (f'_0 + f'_1 z + \cdots + f'_c z^c) \qquad (5.47)$$

The z-transfrom of the linear prefilter of system 4 can now be obtained by substituting eqn. 5.47 into eqn. 5.43 to give

$$F(z) = f'_c + f'_{c-1} z^{-1} + \cdots + f'_0 z^{-c} \qquad (5.48)$$

where f'_c, f'_{c-1}, $\cdots$, f'_0 are now the tap gains of the prefilter here. Consequently, so long as the inequalities of (5.40) remain valid, the partially-inverse prefilter (eqn. 5.43) of system 4 can readily be evaluated from the first c+1 components of the channel sampled impulse response as is described above. Clearly, the linear prefilter here is only slightly more complex than that used in system 3 but is still very much simpler in comparison with the linear prefilter used in system 1.

The near-maximum likelihood detector and the intersymbol interference canceller of system 4 (Fig. 5.5) operate in exactly the same way as those of system 3. Thus, the intersymbol interference canceller operates on the prefilter output samples values (eqn. 5.29) to give, at the detector input, the sample values $r'_{i,0}, r'_{i,1}, \cdots, r'_{i,f}$ (eqn. 5.35). On receiving $r'_{i,f}$, each of the m stored vectors $\{X_{i-1}\}$ (eqn. 5.8) in the detector is expanded into 16 vectors $\{P_i\}$ (eqn. 5.9). The detector then evaluates the cost C_i (by using eqns. 5.11 and 5.37) and the quantity C'_i (by using eqn. 5.38) for each of the 16m expanded vectors $\{P_i\}$. The detected data symbol s'_{i-n} is then taken as the value of x_{i-n} in the vector P_i associated with the smallest value of C'_i, and the detector then proceeds to select the m stored vectors $\{X_i\}$ in exactly the same way as for the detector of system 3.

Thus, system 4 differs from system 3 only in having different tap-gain values $\{f_j\}$ for the prefilter, and so the argument used in system 3 to select the first significant component y'_f for the detection process can be carried over to system 4. That is, at high signal to to noise ratios, the first significant component y'_f in the sampled impulse response of the channel and prefilter should be selected to maximize the quantity d defined by eqn. 5.39. Table 5.3 shows some values of d for system 4 operating over channel A (Table 2.4).

Computer simulation tests have been carried out to determine the tolerance to noise of system 4 operating over channel A (Table 2.4). Various values of f are tested here for each of the values of p shown in Table 5.3, and the simulation results for the system with the best values of f at p = 1 and p = 2 are shown in Fig. 5.7. p+1 is here the number of taps used in the prefilter (Fig. 5.5) and y'_f is the first significant component of the sampled impulse response of the channel and prefilter.

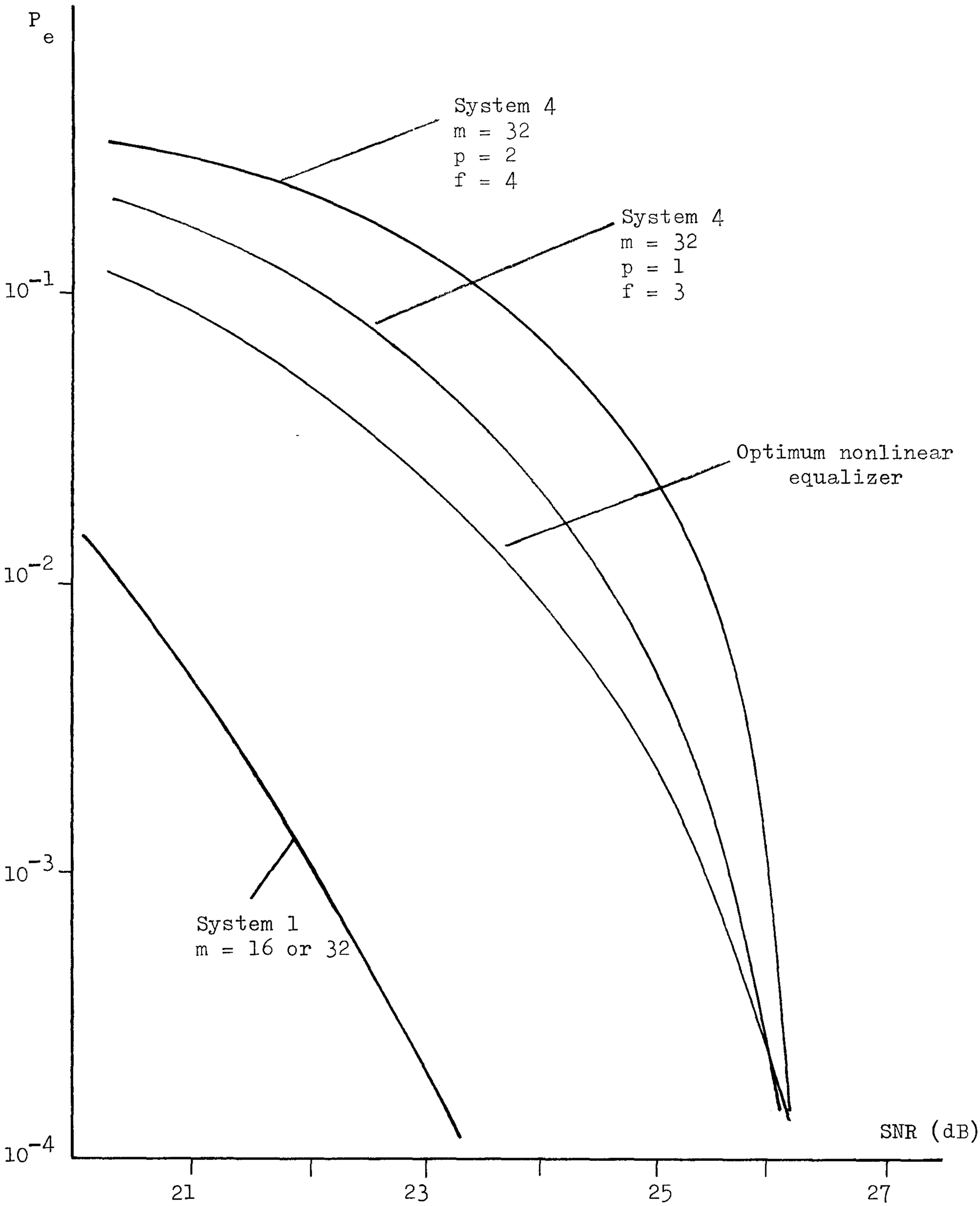


Fig. 5.7 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 4 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16.

The 95% confidence limits of the results shown in Fig. 5.7 are about $\pm$0.5 dB, and the bit error rate and the signal to noise ratio here are as defined by eqns. 5.3 and 5.4 respectively.

Thus, system 4 with p = 1 has the best performance when f = 3, and the system with p = 2 has the best performance when f = 4. By comparing these results with those shown in Fig. 5.6, and from Tables 5.2 and 5.3, it can be seen that any of the systems 3 and 4 that has a larger value of $|y'_f/k_p|$ appears to have a higher tolerance to noise at low signal to noise ratios where the error rates are higher than 10^{-2}, when operating over channel A here. At higher signal to noise ratios where the error rates drop to about 10^{-4}, the best tolerance to noise achieved by system 4 here appears to be better than that achieved by system 3 which may have about the same value of d (eqn. 5.39) but a smaller value of $|y'_f/k_p|$ as compared to the corresponding values of system 4. It therefore appears that, for the values of signal to noise ratios considered here, a better performance over channel A is likely to be achieved by the system having the arrangement of Fig. 5.5 that gives not only a larger value of d but also a larger value of $|y'_f/k_p|$.

Thus, system 4 (with a partially-inverse prefilter) appears to have a slightly better tolerance to noise than that of system 3 at high signal to noise ratios, when operating over channel A here. However, the performance of system 4 still appears to be inferior to that of the optimum nonlinear equalizer as can be seen from Fig. 5.7. One weakness of the (p+1)-tap partially-inverse prefilter used in system 4 is that, it has no control over the magnitudes of the components y'_p $y'_{p+1}, \cdots, y'_\gamma$ although it reduces the magnitudes of the first p components $y'_0, y'_1, \cdots, y'_{p-1}$ of the channel and prefilter sampled impulse response.

This implies that there is now an uncertainty in the magnitude of the first significant component y'_f here. Further modifications to this prefilter are therefore needed and this leads to the development of system 5.

p	f	$\left\|\frac{y'_f}{k_p}\right\|$	$3\sum_{k=0}^{f-1}\left\|\frac{y'_k}{k_p}\right\|$	d (eqn. 5.39)
1	0	0.003	0.000	0.003
	1	0.000	0.009	-0.009
	2	0.124	0.009	0.115
	3	0.434	0.381	0.053
2	0	0.002	0.000	0.002
	1	0.010	0.006	0.004
	2	0.000	0.036	-0.036
	3	0.048	0.036	0.012
	4	0.282	0.180	0.102

Table 5.3 Calculated values of d (eqn. 5.39) for system 4 operating over channel A (Table 2.4). p+1 is the number of taps of the prefilter shown in Fig. 5.5, and k_p is the constant defined by eqn. 5.34.

5.7 System 5

Previous sections have shown that the partially-matched prefilter of system 3 maximizes the magnitude of the first significant component, and that the partially-inverse prefilter of system 4 reduces the magnitudes of the first few components of the channel and prefilter sampled impulse response $y'_0, y'_1, \cdots, y'_\gamma$. Clearly, these two linear prefilters can be connected in cascade to give a linear prefilter that may maximize the magnitude of the first significant component y'_f while at the same time minimizing the magnitudes of the preceeding components here. This is in fact the linear prefilter used in system 5.

Thus, system 5 has the same model of signal processor as those of systems 3 and 4, and is as shown in Fig. 5.5. The linear prefilter here is a (p+1)-tap linear feedforward transversal filter having the z-transform

$$\begin{aligned} F(z) &= f_0 + f_1 z^{-1} + \cdots + f_p z^{-p} \\ &= F_1(z)\, F_2(z) \end{aligned} \tag{5.49}$$

where $\{f_j\}$ are the tap gains of the prefilter, $F_1(z)$ is the z-transform of the partially-matched filter used in system 3, and $F_2(z)$ is the z-transform of the partially-inverse filter used in system 4. Thus, from eqns. 5.27 and 5.48,

$$F_1(z) = y^*_{(p/2)} + y^*_{(p/2)-1} z^{-1} + \cdots + y^*_0 z^{-(p/2)} \tag{5.50}$$

$$\begin{aligned} F_2(z) &= f'_c + f'_{c-1} z^{-1} + \cdots + f'_0 z^{-c} \\ &\simeq z^{-2c} \left(y_0 + y_1 z^{-1} + \cdots + y_c z^{-c} \right)^{-1} \end{aligned} \tag{5.51}$$

where $p = 2c$, y^*_j is the complex conjugate of y_j, and $\{f'_k\}$ are as

given by eqns. 5.45 and 5.46.

The near-maximum likelihood detector and the intersymbol interference canceller of system 5 (Fig. 5.5) operate in exactly the same way as those of system 3 or 4. That is, the intersymbol interference canceller operates on the prefilter output sample values (eqn. 5.29) to give the samples values $\{r'_{i,h}\}$ (eqn. 5.35) which are then fed to the detector. The detector then proceeds with the detection of the data symbols $\{s_i\}$ using the same near-maximum likelihood detection process as that used in each of the systems 3 and 4.

Again, system 5 differs from system 3 or 4 only in having different tap-gain values $\{f_j\}$ for the prefilter, and so the same argument used in system 3 or 4 to select the first significant component y'_f for the detection process can be carried over to system 5. That is, at high signal to noise ratios, the component y'_f (or the value of f) should be selected to maximize the quantity d given by eqn. 5.39, where the value of d basically gives a measure of the relative magnitude of the first significant component to the preceeding components of the channel and prefilter sampled impulse response $y'_0, y'_1, \cdots, y'_\gamma$. Table 5.4 shows some values of d for system 5 operating over channel A (Table 2.4).

Computer simulation tests have been carried out to determine the tolerance to noise of system 5 operating over channel A (Table 2.4). Various values of f have been tested here for each of the two values of p shown in Table 5.4, and the results for the system with best values of f at p = 2 and p = 4 are shown in Fig. 5.8. p+1 is, of course, the number of taps used in the prefilter here (Fig. 5.5). The 95% confidence limits of the simulation results shown in Fig. 5.8 are about ± 0.5 dB. The definitions of the bit error rate and the signal to noise ratio here are as given by eqns. 5.3 and 5.4 respectively.

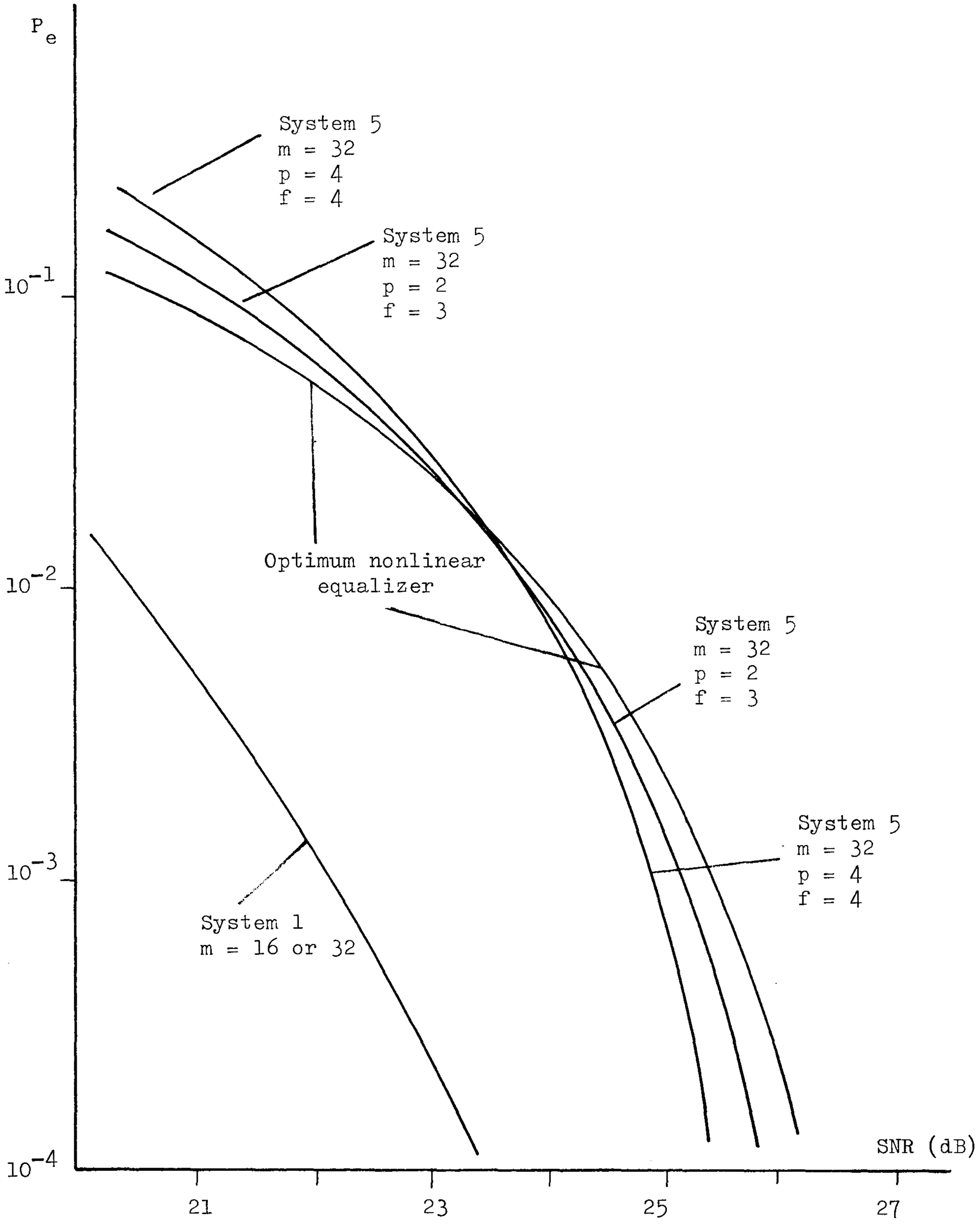


Fig. 5.8 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 5 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16.

p	f	$\left\|\frac{y'_f}{k_p}\right\|$	$3\sum_{k=0}^{f-1}\left\|\frac{y'_k}{k_p}\right\|$	d (eqn. 5.39)
2	0	0.003	0.000	0.003
	1	0.000	0.009	-0.009
	2	0.125	0.009	0.116
	3	0.454	0.384	0.070
4	0	0.002	0.000	0.002
	1	0.011	0.006	0.005
	2	0.004	0.039	-0.035
	3	0.052	0.051	0.001
	4	0.321	0.207	0.114

Table 5.4 Calculated values of d (eqn. 5.39) for system 5 operating over channel A (Table 2.4). p+1 is the number of taps of the prefilter shown in Fig. 5.5, and k_p is the constant defined by eqn. 5.34.

It can be seen from Table 5.4 and Fig. 5.8 that system 5 with p = 2 and f = 3 has a larger value of $\left|y'_f/k_p\right|$ and a higher tolerance to noise at low signal to noise ratios in comparison with those of the system with p = 4 and f = 4. The latter, however, has a larger value of d and a higher tolerance to noise at high signal to noise ratios when the error rates drop below 10^{-3}. These behaviours are in agreement with the results obtained previously for any of the systems 2, 3, and 4.

As Figs. 5.4, 5.6, 5.7, and 5.8 show, system 5 appears to have a better tolerance to Gaussian noise than those of the systems 2, 3, and 4, when operating over channel A here. At an error rate of 10^{-4}, the tolerance to noise of system 5 here is about 0.5 dB better than that of the optimum nonlinear equalizer. The performance of this system is, however, inferior to that of the equalizer at low signal to noise ratios. Since the number of stored vectors (being denoted as m in Fig. 5.8) used in the detection process of system 5 here is 32, the number of operations per data symbol involved in this system is very large as compared to that involved in the equalizer. Consequently, system 5 does not appear to be a promising system as it involves a much more complex detection process while having only a very small improvement in tolerance to noise over the optimum nonlinear equalizer at high signal to noise ratios.

It can be seen from the study of systems 3, 4, and 5 operating over channel A (Table 2.4) that a higher tolerance to noise at high signal to noise ratios may be achieved by the system having a larger value of d (eqn. 5.39) and $|y'_f/k_p|$ (eqns. 5.32 and 5.34). The value of d here gives a measure of the relative magnitude of the first significant component y'_f to the preceeding components of the channel and prefilter sampled impulse response, and k_p is a constant such that the noise variance remains unchanged at the output of the prefilter used in the system here. However, the linear prefilter here, being derived from the first few components of the channel sampled impulse response, appears to modify only very slightly the relative magnitude of the first significant component to the preceeding components of the channel sampled impulse response. Consequently, only a small improvement in tolerance to noise over system 2 may be gained here. The system that uses a short linear prefilter therefore does not appear to be promising and is thus not given any further consideration here.

5.8 System 6

Previous studies have shown that the use of a short linear prefilter at the detector input could only give a limited gain in tolerance to Gaussian noise when operating over channel A (Table 2.4). The alternative approach of discarding entirely the linear prefilter and using a modified near-maximum likelihood detection process is now considered.

It is recalled that the basic mechanism of system 2 is to ignore the first f components of any received signal element until the (f+1)th component (the first significant component) is received, when all components of the signal element are taken account of in the detection process. The ignored components here are referred to as the intersymbol interference components and they may also be considered as the additional noise components added to the system. It has been discussed in section 5.4 that, at high signal to noise ratios, the net gain in the system performance is likely to be larger when the relative magnitudes of the intersymbol interference components in the first significant component are smaller. It therefore appears that, a higher tolerance to noise may be achieved here if the detection process of system 2 is modified to reduce the magnitudes of the intersymbol interference components mentioned above. This leads to the development of system 6.

The model of the signal processor of system 6 is shown in Fig. 5.9 where the data symbol s_{i-n} is detected after the receipt of the sample value $r_{i+\alpha}$. The sample values $\{r_{i+h}\}$ are as defined by eqn. 5.1, and α is an appropriate positive integer to be discussed shortly.

Thus, after the receipt of $r_{i+\alpha}$, the intersymbol interference canceller (Fig. 5.9) removes from the samples $r_i, r_{i+1}, \cdots, r_{i+\alpha}$ detected values of all components involving data symbols whose final

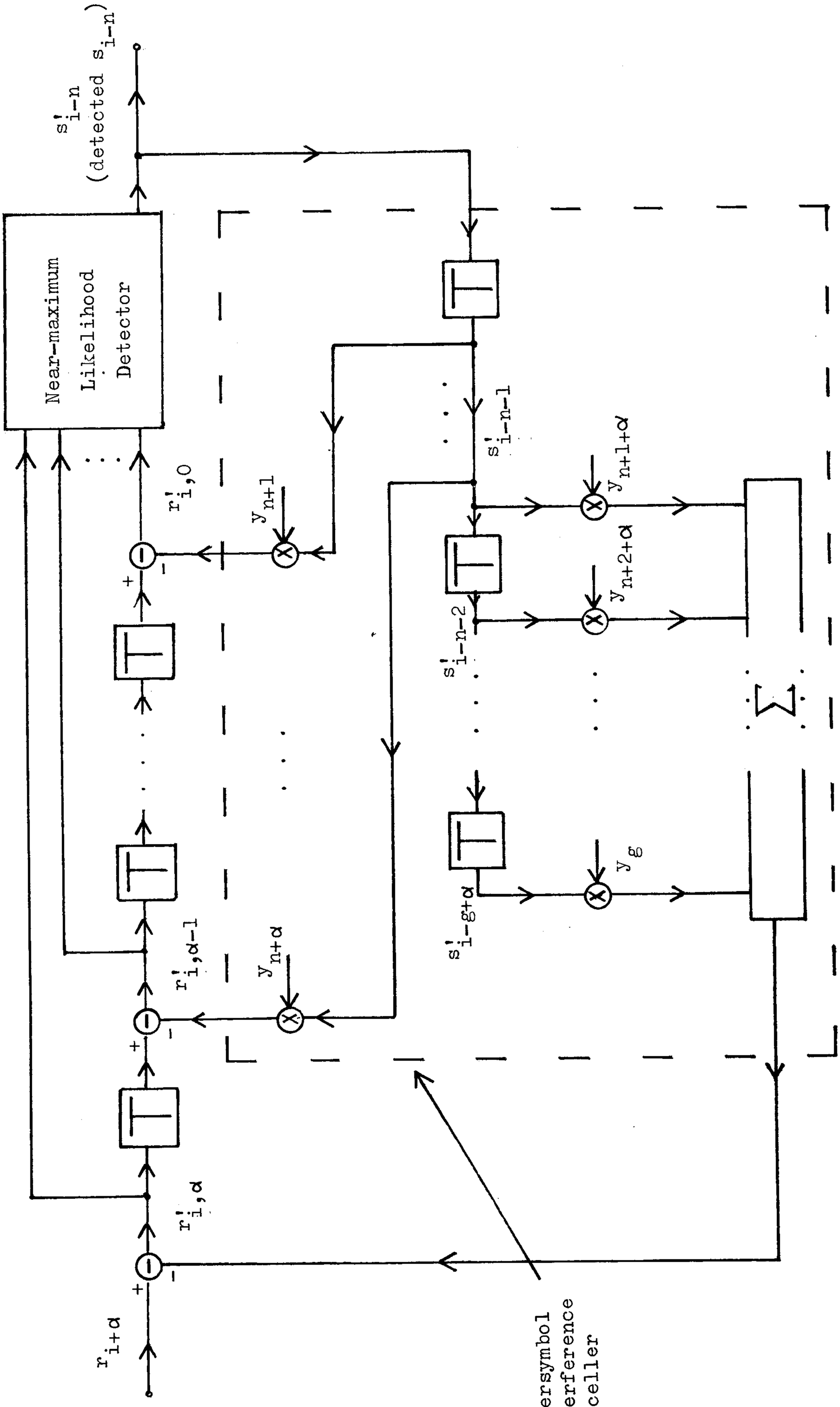


Fig. 5.9 Signal processor of any of the systems 6, 7, and 8. $\alpha > f$, and y_f is the first significant component of the channel sampled impulse response.

detected values $s'_{i-n-1}, s'_{i-n-2}, \cdots, s'_{i-g}$ have already been determined. That is, the intersymbol interference canceller operates on $r_i, r_{i+1}, \cdots, r_{i+\alpha}$ to give the following complex-valued quantities

$$\begin{aligned}
r'_{i,0} &= r_i - \sum_{h=n+1}^{g} s'_{i-h} y_h \\
r'_{i,1} &= r_{i+1} - \sum_{h=n+2}^{g} s'_{i+1-h} y_h \\
&\vdots \\
r'_{i,\alpha} &= r_{i+\alpha} - \sum_{h=n+\alpha+1}^{g} s'_{i+\alpha-h} y_h
\end{aligned} \tag{5.52}$$

where $y_0, y_1, \cdots, y_g$ are the g+1 components of the channel sampled impulse response. Assuming for the moment that $s'_{i-h} = s_{i-h}$ for $h = n+1, n+2, \cdots, g$, eqn. 5.52 reduces to

$$\begin{aligned}
r'_{i,0} &= \sum_{h=0}^{n} s_{i-h} y_h + w_i \\
r'_{i,1} &= \sum_{h=0}^{n+1} s_{i+1-h} y_h + w_{i+1} \\
&\vdots \\
r'_{i,\alpha} &= \sum_{h=0}^{n+\alpha} s_{i+\alpha-h} y_h + w_{i+\alpha}
\end{aligned} \tag{5.53}$$

where the real and imaginary parts of the noise components $\{w_i\}$ here are statistically independent Gaussian random variables with zero mean and variance σ^2.

The detector (Fig. 5.9) here operates on its input sample values $r'_{i,0}, r'_{i,1}, \cdots, r'_{i,\alpha}$ to give at its output the detected data symbol s'_{i-n}, so that there is here a delay in detection of $n+\alpha$ sampling intervals. This detection process is now described as follows. Just prior to the receipt of the sample $r_{i+\alpha}$, the detector of system 6 holds in store m n-component vectors $\{X_{i-1}\}$ (eqn. 5.8) together with their associated costs $\{C_{i-1}\}$, where the costs here are as defined by eqns. 5.10, 5.11, and 5.16. On receiving $r_{i+\alpha}$, the detector expands each vector X_{i-1} into the corresponding 16 vectors $\{P_i\}$ (eqn. 5.9) having the 16 possible values of x_i and it evaluates the cost C_i for each of the 16m expanded vectors $\{P_i\}$ using eqns. 5.11 and 5.16. The detection process so far is the same as that of system 2. The detector of system 6 next expands each vector P_i into 16^{λ} $(n+1+\lambda)$-component vectors $\{P'_i\}$ where

$$P'_i = \left[\begin{array}{ccccccc} x_{i-n} & x_{i-n+1} & \cdot\cdot & x_i & x_{i+1} & \cdot\cdot & x_{i+\lambda} \end{array} \right] \quad (5.54)$$

and $\lambda > 0$, bearing in mind that x_j here may have any of the 16 possible values of the 16-point QAM data symbol s_j for any possible value of j. In system 6, the value of λ is related to α (Fig. 5.9) by

$$\alpha = \lambda + f \quad (5.55)$$

where y_f is the first significant component of the channel sampled impulse response here. The first n+1 components of P'_i in eqn. 5.54 are as in the original vector P_i (eqn. 5.9) and the last λ components $x_{i+1}, x_{i+2}, \cdots, x_{i+\lambda}$ have the 16^{λ} possible combinations of the values of $s_{i+1}, s_{i+2}, \cdots, s_{i+\lambda}$. A schematic diagram showing the expansion of 2 vectors $\{X_{i-1}\}$ to $2x16^2$ vectors $\{P'_i\}$ is given in Fig. 5.10. There are now altogether $16^{\lambda+1}m$ vectors $\{P'_i\}$ each of which has the same cost C_i as that of its original vector P_i. The detector then evaluates

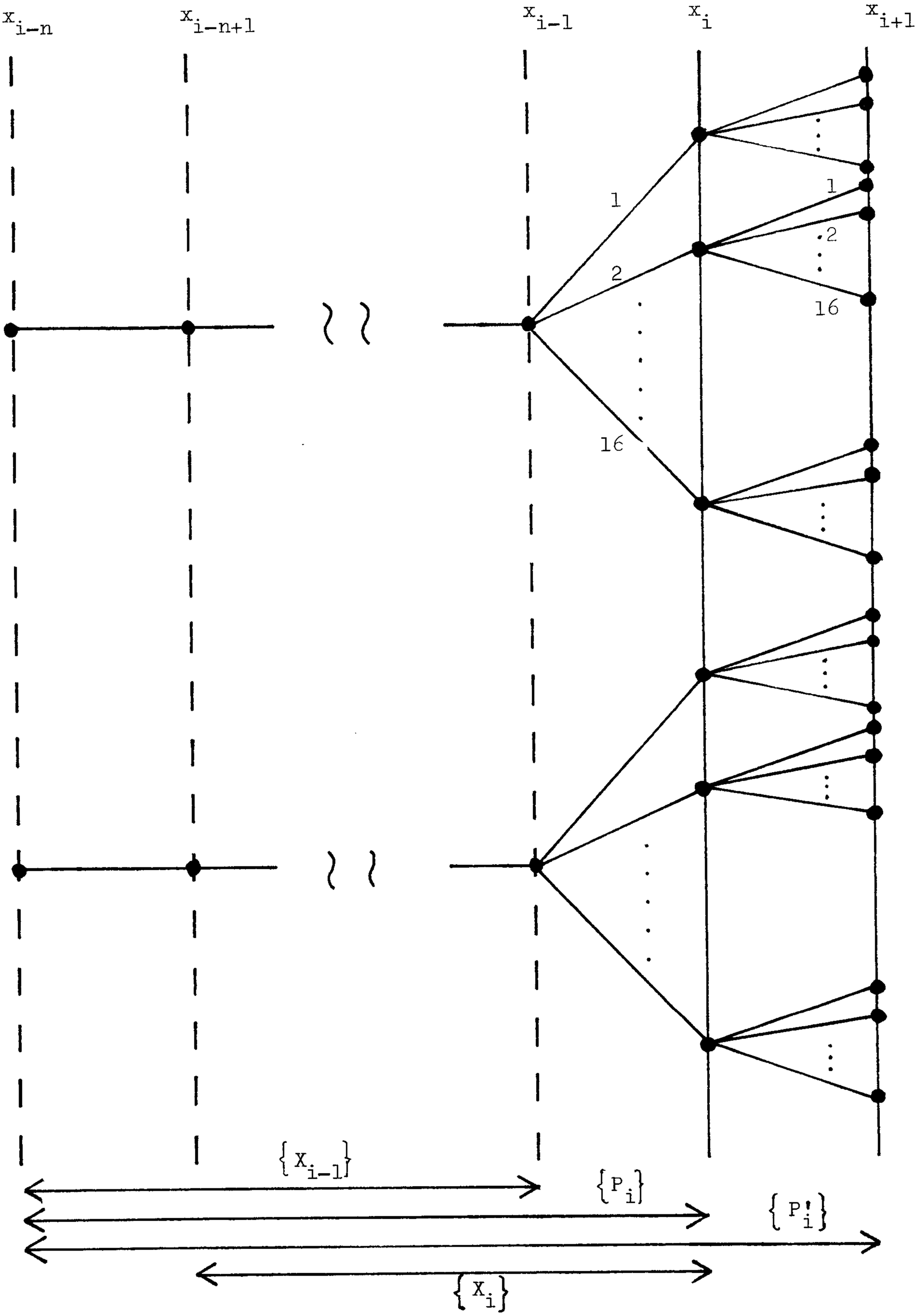


Fig. 5.10 Schmatic diagram showing the expansion of vectors from $m\{X_{i-1}\}$ to $16^{\lambda+1} m\{P_i'\}$ in the detection process of system 6. $m = 2$, $\lambda = 1$ in this example. The components $\{x_j\}$ are here represented by the symbol ● .

the quantity C_i' for each of the $16^{\lambda+1}m$ vectors $\{P_i'\}$, where

$$C_i' = C_i + \sum_{h=1}^{\alpha} \left| r_{i,h}' - \sum_{j=0}^{n+h} x_{i+h-j} y_j \right|^2 \qquad (5.56)$$

$x_k = 0$ for $k \leq 0$, $k > i+\lambda$ (see eqn. 5.54), and the $\{r_{i,h}'\}$ are as defined by eqn. 5.52. The $\{y_j\}$ in eqn. 5.56 are, of course, the components of the channel sampled impulse response and α is the positive integer given in eqn. 5.55. The significance of the value of C_i' here will be further discussed shortly. Having evaluated the value of C_i' for each of the $16^{\lambda+1}m$ vectors $\{P_i'\}$, the detector then selects, for each vector P_i, the vector P_i' associated with the smallest value of C_i' to give a total of 16m selected vectors $\{P_i'\}$. The remaining vectors of $\{P_i'\}$ are then discarded from further consideration. Thus, each of the 16m selected vectors $\{P_i'\}$ here has a different set of values for the components $x_{i-n}, x_{i-n+1}, \cdots, x_i$ so long as each of the m original vectors $\{P_i\}$ has a different set of values for its components. This necessarily prevents the stored vectors $\{X_i\}$ (to be selected shortly) from merging as can be seen later. Having selected the 16m vectors $\{P_i'\}$ described above, the detected data symbol s_{i-n}' is then taken as the value of x_{i-n} in the vector P_i' associated with the smallest value of C_i'. All vectors $\{P_i'\}$ for which $x_{i-n} \neq s_{i-n}'$ are now discarded. The first component x_{i-n} and the last λ components $x_{i+1}, x_{i+2}, \cdots x_{i+\lambda}$ of each of the remaining vectors $\{P_i'\}$ are omitted to give the corresponding n-component vectors $\{X_i\}$. Each vector X_i is, of course, associated with the same C_i and C_i' as those of the corresponding P_i', and since each of the vectors $\{P_i'\}$ here has a differect set of values for the components $x_{i-n+1}, x_{i-n+2}, \cdots, x_i$, the vectors $\{X_i\}$ here must all be different. The detector now selects the m vectors $\{X_i\}$ associated with the smallest values of $\{C_i'\}$ in the same way as is carried out by the detector of

system 2. That is, the first selected vector is the vector X_i associated with the smallest value of C_i'. The second vector is selected with the constraint that only the 16 vectors $\{X_i\}$ originating from the previous first (best) vector X_{i-1} are available for the selection here. The next m-2 vectors are selected from the remaining vectors $\{X_i\}$ associated the smallest values of $\{C_i'\}$. These m selected vectors $\{X_i\}$, together with their associated costs $\{C_i\}$ are then stored and the detector is now ready for the next detection process. It should be noted that, since all the vectors $\{X_i\}$ available for the selection of the m finally stored vectors $\{X_i\}$ just mentioned are different, all the m finally stored vectors here must also be different and are hence prevented from merging.

The significance of the quantity C_i' (eqn. 5.56) used in the detection process of system 6 is now discussed. By substituting the values of $\{r'_{i,h}\}$ from eqn. 5.53 into eqn. 5.56 and bearing in mind that $x_k = 0$ for $k \leq 0$, $k > i+\lambda$, the value of C_i' now reduces to

$$C_i' = C_i + \sum_{h=1}^{\alpha} \left| \sum_{j=0}^{n+h} (s_{i+h-j} - x_{i+h-j}) y_j + w_{i+h} \right|^2$$

$$= C_i + \sum_{h=1}^{\lambda+f} \left| \sum_{j=0}^{n+h} (s_{i+h-j} - x_{i+h-j}) y_j + w_{i+h} \right|^2$$

$$= C_i + \sum_{k=1}^{\lambda} \left| \sum_{j=0}^{n+k} (s_{i+k-j} - x_{i+k-j}) y_j + w_{i+k} \right|^2$$

$$+ \sum_{k=\lambda+1}^{\lambda+f} \left| \sum_{j=k-\lambda}^{n+k} (s_{i+k-j} - x_{i+k-j}) y_j + I_{k-\lambda} + w_{i+k} \right|^2 \qquad (5.57)$$

and $$I_h = \sum_{j=0}^{h-1} s_{i+\lambda+h-j} y_j \qquad (5.58)$$

for $h = 1, 2, \cdots, f$ where λ, α are as given by eqns. 5.54, 5.55, and 5.56, and the $\{I'_h\}$ are the intersymbol interference components caused by ignoring the presence of the f data symbols $s_{i+\lambda+1}, s_{i+\lambda+2}, \cdots, s_{i+\lambda+f}$ in $r'_{i,\lambda+1}, r'_{i,\lambda+2}, \cdots, r'_{i,\lambda+f}$ (eqn. 5.53) in the computation of the quantity C'_i here. At high signal to noise ratios when the error rates are low, it is very likely that the vector X_i whose components are all correct (that is, $x_h = s_h$ for $h = i-n+1, i-n+2, \cdots, i$) is available for the selection process that selects the m finally stored vectors $\{X_i\}$ associated with the smallest values $\{C'_i\}$. Intuitively, the probability of error in detection is now likely to be determined to a large extent by the probability of discarding this vector of X_i. The value of C'_i for this vector of X_i is given by

$$C'_i = C_i + \sum_{k=1}^{\lambda} \left| \sum_{j=0}^{k} (s_{i+k-j} - x_{i+k-j}) y_j + w_{i+k} \right|^2 + \sum_{k=\lambda+1}^{\lambda+f} \left| \sum_{j+k-\lambda}^{k} (s_{i+k-j} - x_{i+k-j}) y_j + I_{k-\lambda} + w_{i+k} \right|^2 \qquad (5.59)$$

where all the components involving $s_{i-n}, s_{i-n+1}, \cdots, s_{i-1}$ in C'_i (eqn. 5.57) have been removed here. Furthermore, since all the possible values of $s_{i+1}, s_{i+2}, \cdots, s_{i+\lambda}$ are considered for the components $x_{i+1}, x_{i+2}, \cdots, x_{i+\lambda}$ in the detection process here, the value of C'_i in eqn. 5.59 may be selected to become

$$C'_i = C_i + \sum_{k=1}^{\lambda} \left| (s_i - x_i) y_k + w_{i+k} \right|^2 + \sum_{k=\lambda+1}^{\lambda+f} \left| (s_i - x_i) y_k + I_{k-\lambda} + w_{i+k} \right|^2 \qquad (5.60)$$

where all the components involving $s_{i+1}, s_{i+2}, \cdots, s_{i+\lambda}$ in C'_i (eqn. 5.59)

have been removed here. The second term of eqn. 5.60 suggests that, the arrangement of system 6 that uses a larger value of λ (eqn. 5.54) has a value of C_i' that includes more components of the signal element associated with s_i, and must therefore have a higher tolerance to noise. Furthermore, since the third term of eqn. 5.60 resembles the second term of eqn. 5.17 (for system 2), the argument used in system 2 to select the value of f for use at high signal to noise ratios may be carried over to system 6. That is, at high signal to noise ratios and for a given value of λ., the value of f to be used in system 6 should be selected to maximize the quantity

$$d = |y_{\lambda+f}| - 3 \sum_{k=0}^{f-1} |y_k| \qquad (5.61)$$

where $y_0, y_1, \cdots, y_g$ are, of course, the components of the channel sampled impulse response, and $|y_j|$ is the modulus of y_j.

The drawback of system 6 is that, the number of operations per data symbol involved here increases sharply with the value of λ. In particular, a measurement of the quantity C_i' (eqn. 5.56) is required for each of the $16^{\lambda+1}m$ expanded vectors $\{P_i'\}$ (eqn. 5.54), where m is the number of stored vectors $\{X_i\}$ (eqn. 5.8) here. Even if $\lambda = 1$, the total number of measurements of $\{C_i'\}$ involved in system 6 is 256m which is a very large number as compared to that involved in system 2 where the number of measurements of $\{C_i'\}$ is 16m. Consequently, only the value of $\lambda = 1$ is considered for system 6 here.

Computer simulation tests have been carried out to determine the tolerance to noise of system 6 with $\lambda = 1$ operating over channel A (Table 2.4), and the results are shown in Fig. 5.11. The 95% confidence limits of these results are about ± 0.5 dB, and the definitions of the bit error rate and the signal to noise ratio here are as given

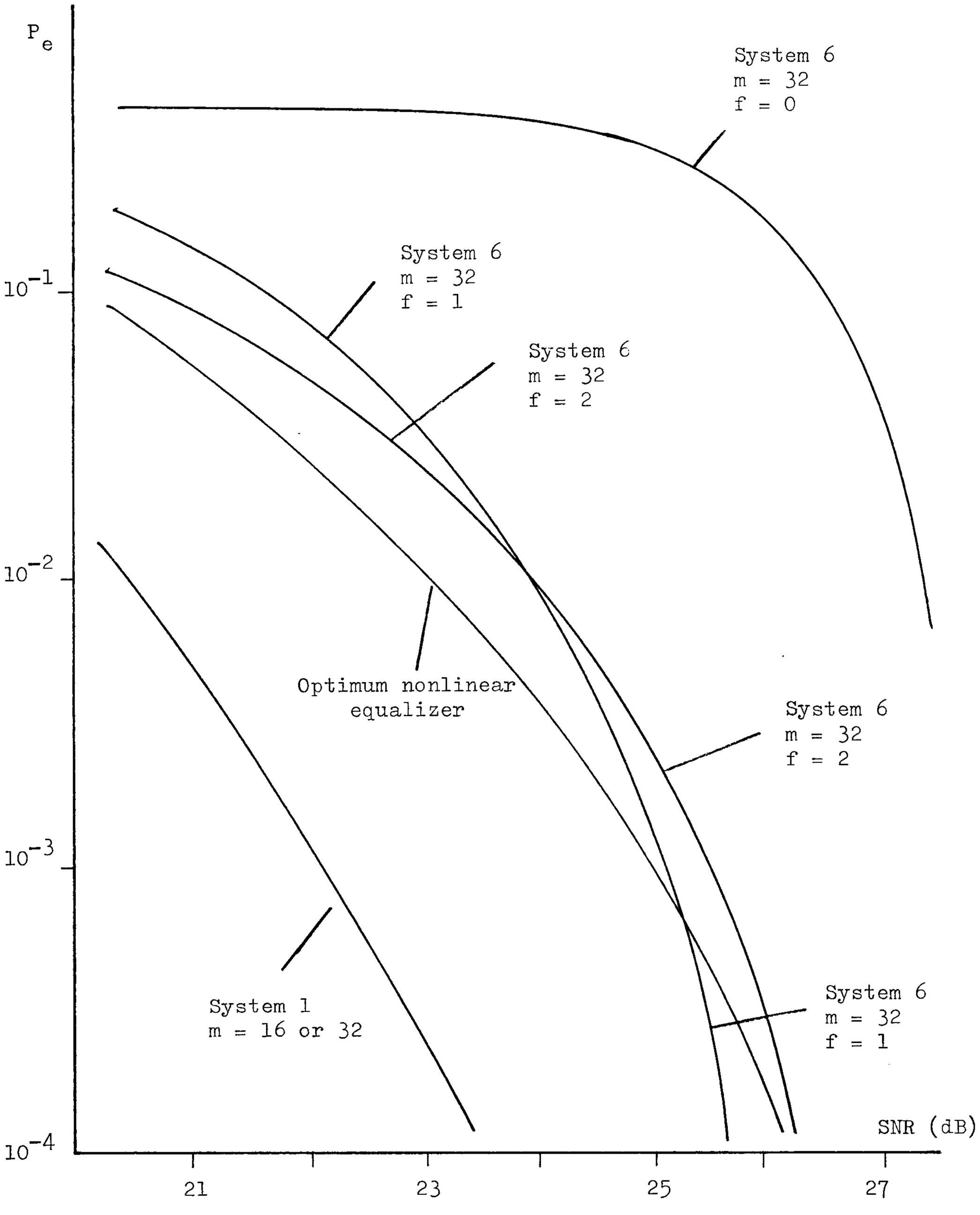


Fig. 5.11 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 6 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16.

by eqns. 5.3 and 5.4 respectively. Table 5.5 also shows the values of the quantity d defined by eqn. 5.61 for the arrangements of system 6 tested in Fig. 5.11.

It can be seen from Table 5.5 and Fig. 5.11 that, system 6 with the arrangement that has the largest value of d (eqn. 5.61) has the best performance at high signal to noise ratios, and the system with the arrangement that has the largest value of $|y_{\lambda+f}|$ (λ being equal to 1 here) has the best performance at low signal to noise ratios. These behaviours are in agreement with the results obtained previously for each of the systems 2, 3, 4, and 5. It can be seen from Figs. 5.4 and 5.11 that, system 6 with $\lambda = 1$ is able to give a better tolerance to noise than that of system 2 when operating over channel A here. However, the improvement in tolerance to noise of system 6 with $\lambda = 1$ over the optimum nonlinear equalizer (section 2.6.3) is only about 0.5 dB at an error rate of 10^{-4}, as can be seen from Fig. 5.11. The improvement can be expected to be larger if a larger value of λ is used in system 6, but an even more excessive amount of operations is now involved in the system. Consequently, system 6 does not appear to be a promising system, and further modifications to this system are therefore needed. Clearly, the modifications should aim at reducing the number of operations while keeping the basic idea of considering as many components of the received signal elements as possible in the detection process. This leads to the development of system 7.

f	$\lvert y_f \rvert$	$3 \sum_{k=0}^{f-1} \lvert y_k \rvert$	d (eqn. 5.25)
0	0.186	0.000	0.186
1	0.492	0.075	0.417
2	0.533	0.633	-0.100

Table 5.5 Calculated values of d (eqn. 5.61) for system 6 with $\lambda = 1$ operating over channel A (Table 2.4).

5.9 System 7

In the detection process of system 6, each of the m stored vectors $\{X_{i-1}\}$ (eqn. 5.8) is expanded into $16^{\lambda+1}$ vectors of $\{P'_i\}$ (eqn. 5.54) having all the $16^{\lambda+1}$ possible combinations of the values for the $\lambda+1$ components $x_i, x_{i+1}, \cdots, x_{i+\lambda}$. However, only m vectors of $\{P'_i\}$ are finally selected and converted to the corresponding m stored vectors $\{X_i\}$. Since m is usually much smaller than $16^{\lambda+1}$, many of the $16^{\lambda+1}$ vectors $\{P'_i\}$ originating from a vector of X_{i-1} will not be selected here. Consequently, it appears that, each vector X_{i-1} may be expanded into much fewer (that is, $\ll 16^{\lambda+1}$) vectors of $\{P'_i\}$ in the detection process while having only a very small reduction in tolerance to noise. This is the basis of system 7.

Thus, system 7 is a modification of system 6 in that, each vector X_{i-1} is expanded into a small number ($\ll 16^{\lambda+1}$) of vectors $\{P'_i\}$ here by the use of a linear transversal filter and a threshold device. The basic model of the signal processor here is the same as that shown in Fig. 5.9. The intersymbol interference canceller in Fig. 5.9 operates on the received sample values $r_i, r_{i+1}, \cdots, r_{i+\alpha}$ (eqn. 5.1) to give, at the detector input, the sample values $r'_{i,0}, r'_{i,1}, \cdots, r'_{i,\alpha}$ (eqn. 5.52), as is described before for system 6. The detector (Fig. 5.9) here operates on its input sample values $\{r'_{i,h}\}$ (eqn. 5.52) to give at its output the detected data symbol s'_{i-n}, and this is described as below.

Just prior to the receipt of the samples $\{r'_{i,h}\}$, the detector of system 7 holds in store m n-component vectors $\{X_{i-1}\}$ (eqn. 5.8) together with the associated costs $\{C_{i-1}\}$ (eqns. 5.10, 5.11, and 5.16). On receiving $\{r'_{i,h}\}$, each of the m stored vectors $\{X_{i-1}\}$ is expanded into m' vectors of $\{P'_i\}$ (eqn. 5.54), where m' is a small number to

be discussed shortly. The first n components of these m' vectors $\{P'_i\}$ are as in the original vector X_{i-1}, and the last λ+1 components x_i, x_{i+1}, $\cdots$, $x_{i+\lambda}$ are derived, in turn, as follows. For each of the original vector X_{i-1}, the detector first derives m'_0 values for the component x_i, to give the corresponding m'_0 (n+1)-component expanded vectors $\{P_i\}$ (eqn. 5.9). For each of these m'_0 expanded vectors $\{P_i\}$ (originating from the same X_{i-1} and having the m'_0 values of x_i just derived), the detector next derives m'_1 values for the component x_{i+1} to give a further m'_1 (n+2)-component expanded vectors. The detector continues to operate in this way to derive, in turn, m'_h values for the component x_{i+h}, for h = 2, 3, .. , λ. Consequently, a total of m' (n+λ+1)-component expanded vectors $\{P'_i\}$ (eqn. 5.54) are derived here for each of the m stored vectors $\{X_{i-1}\}$, where

$$m' = m'_0 \, m'_1 \cdots m'_\lambda \qquad (5.62)$$

An example of the operations just described is illustrated by the schematic diagram shown in Fig. 5.12 where a total of 16 vectors $\{P'_i\}$ are expanded from each of the 2 vectors $\{X_{i-1}\}$ here. The derivation of the m'_h values for the component x_{i+h} ($0 \leq h \leq \lambda$) will be described in details later. Thus, having derived the m' vectors $\{P'_i\}$ for each of the m stored vectors $\{X_{i-1}\}$, the detector of system 7 then proceeds to evaluate the costs $\{C_i\}$ (by using eqns. 5.11 and 5.16) and the quantities $\{C'_i\}$ (by using eqn. 5.56) for the (m)(m') expanded vectors $\{P'_i\}$ here. It can be seen from eqns. 5.11 and 5.16 that, those vectors $\{P'_i\}$ originating from the same (n+1)-component vector P_i here (eqn. 5.9 and Fig. 5.12) are associated with the same cost as this vector P_i. Thus, having evaluated the (m)(m'_0) values of $\{C_i\}$ and the (m)(m') values of $\{C'_i\}$, the detector then selects, for each of the (m)(m'_0) vectors $\{P_i\}$, the vector P'_i

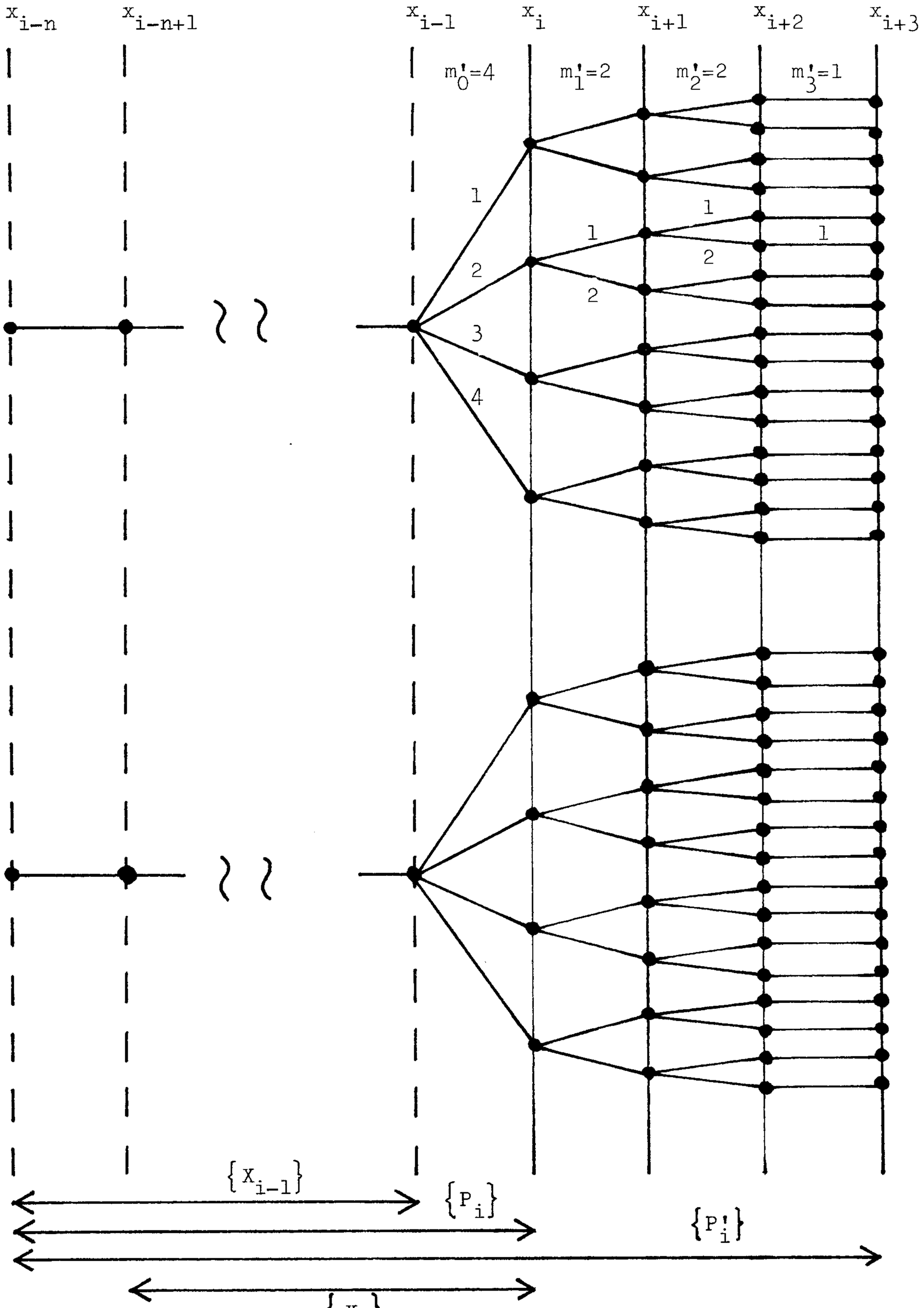


Fig. 5.12 Schematic diagram showing the expansion of vectors from $m\{X_{i-1}\}$ to $(m)(m')\{P'_i\}$ in the detection process of system 7. m' is as defined by eqn. 5.62. In this example, m = 2.

that has the smallest value of C_i', to give a total of $(m)(m_0')$ selected vectors $\{P_i'\}$. The remaining (unselected) vectors $\{P_i'\}$ are then discarded from further consideration. The detected data symbol s_{i-n}' is then taken to have the value of x_{i-n} in the vector P_i' associated with the smallest C_i'. All vectors $\{P_i'\}$ for which $x_{i-n} \neq s_{i-n}$ are now further discarded. The first component x_{i-n} and the last λ components $x_{i+1}, x_{i+2}, \cdots, x_{i+\lambda}$ of each of the remaining vectors $\{P_i'\}$ are omitted to give the corresponding n-component vectors $\{X_i\}$. Each vector X_i here is, of course, associated with the same C_i and C_i' as those of the corresponding vector P_i'. The detector then proceeds with the selection of the m vectors $\{X_i\}$ associated with the smallest $\{C_i'\}$ exactly as for the detector of system 6. These m selected vectors $\{X_i\}$ together with their associated costs $\{C_i\}$ are then stored and the detector is now ready for the next detection process.

Thus, system 7 basically operates in the same way as system 6 except that a selection process is involved here to derive just m' possible vectors $\{P_i'\}$ (instead of using directly all the $16^{\lambda+1}$ possible vectors $\{P_i'\}$) for the detection process, bearing in mind that the value of m' is much smaller than $16^{\lambda+1}$. This selection process involves the derivations of m_h' values for the component x_{i+h} in P_i', for h = 0, 1, $\cdots$, λ. The derivation of the m_h' values for the component x_{i+h} here is carried out by using a (p+1)-tap linear transversal filter and a threshold device, and this derivation process is now described as follows.

The detector first removes from the p+1 samples $r_{i,h}', r_{i,h+1}', \cdots, r_{i,h+p}'$ (eqn. 5.52) detected values of all components involving the data symbols $s_{i-n}, s_{i-n+1}, \cdots, s_{i+h-1}$ whose values have already been determined (temporarily) as the values of the corresponding components $x_{i-n}, x_{i-n+1}, \cdots, x_{i+h-1}$. That is, the detector operates on $r_{i,h}', r_{i,h+1}', \cdots, r_{i,h+p}'$ to give the following complex-valued quantities

$$e'_0 = r'_{i,h} - \sum_{k=1}^{n+h} x_{i+h-k} y_k$$

$$e'_1 = r'_{i,h+1} - \sum_{k=1}^{n+h} x_{i+h-k} y_{k+1}$$

.

.

.

$$e'_p = r'_{i,h+p} - \sum_{k=1}^{n+h} x_{i+h-k} y_{k+p} \qquad (5.63)$$

where $\{y_j\}$ are the components of the channel sampled impulse response. The detector then operates on these sample values (eqn. 5.63), using the (p+1)-tap linear transversal filter shown in Fig. 5.13, to give the complex-valued quantity

$$x'_{i+h} = \frac{1}{k_p^2} \left(e'_0 y_0^* + e'_1 y_1^* + \cdots + e'_p y_p^* \right) \qquad (5.64)$$

where $$k_p^2 = \sum_{k=0}^{p} |y_k|^2 \qquad (5.65)$$

and y_j^* is the complex conjugate of y_j. The value of p here is related to the values of α and λ by

$$\alpha = \lambda + p \qquad (5.66)$$

where α and λ are as defined in Fig. 5.9 and eqn. 5.54 respectively. The detector now proceeds with the selection process that operates on the value of x'_{i+h} (eqn. 5.64) to select m'_h possible values for the component x_{i+h} (eqn. 5.54). In this selection process, the detector assumes that the values of $r'_{i,h}, r'_{i,h+1}, \cdots, r'_{i,h+p}$ in eqn. 5.63 are as given by eqn. 5.53, and that $x_j = s_j$ for $j = i-n, i-n+1, \cdots, i+h-1$, so that eqn. 5.63 now becomes

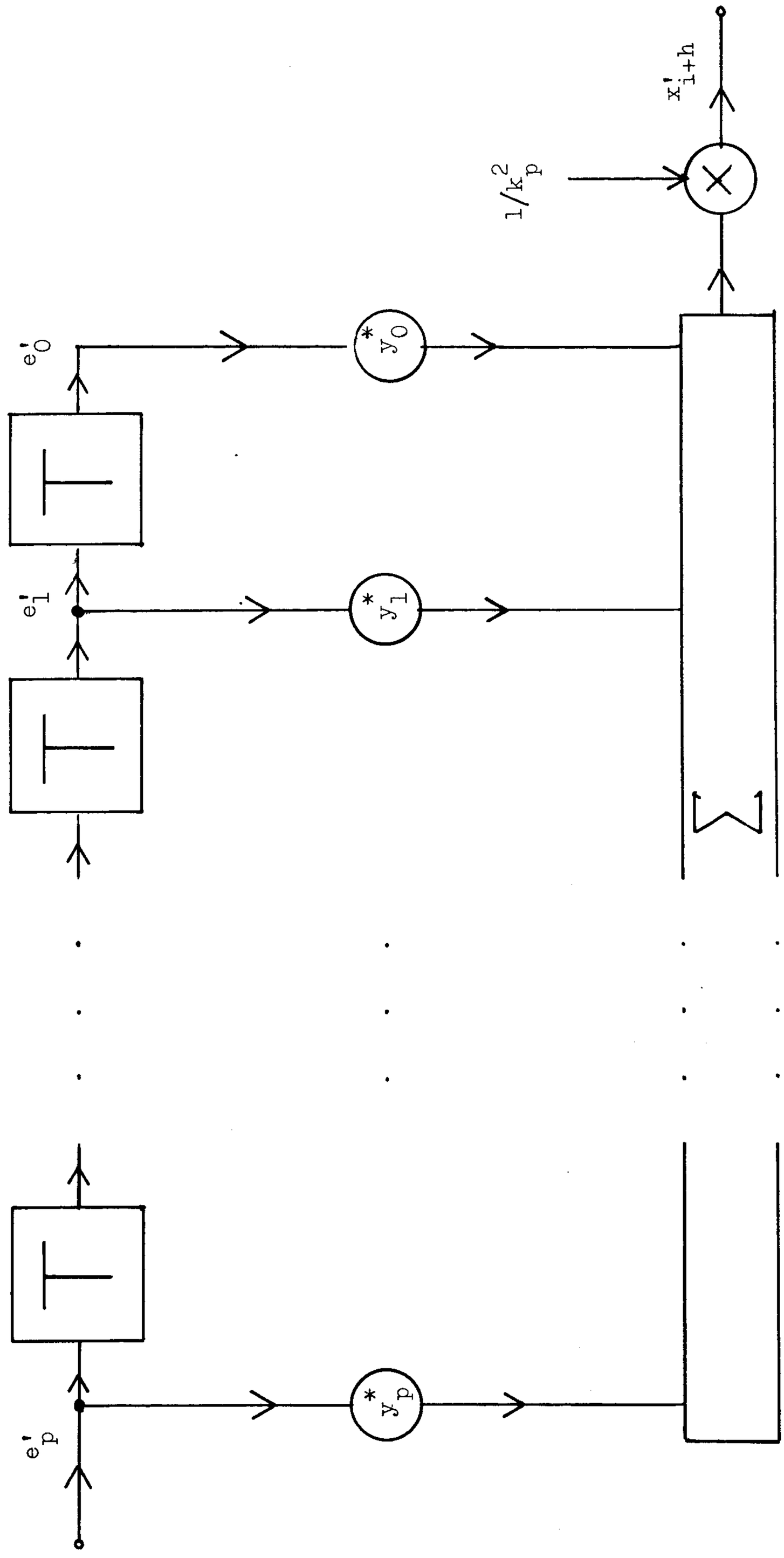


Fig. 5.13 Linear transversal filter used in system 7 to evaluate the quantities $\{x'_{i+h}\}$ in deriving the possible values for the corresponding $\{x_{i+h}\}$ in the expanded vectors $\{P'_i\}$ (eqn. 5.54), for h = 0, 1, .. , ν.

$$e'_0 = s_{i+h}y_0 + w_{i+h}$$

$$e'_1 = s_{i+h}y_1 + s_{i+h+1}y_0 + w_{i+h+1}$$

$$\vdots$$

$$e'_p = s_{i+h}y_p + s_{i+h+1}y_{p-1} + \cdots + s_{i+h+p}y_0 + w_{i+h+p} \qquad (5.67)$$

where $\{w_j\}$ are the noise components whose real and imaginary parts are statistically independent Gaussian random variables with zero mean and variance σ^2. Eqn. 5.64 now reduces to

$$x'_{i+h} = s_{i+h} + s_{i+h+1}a'_1 + \cdots + s_{i+h+p}a'_p + w'_{i+h}$$

$$= s_{i+h} + J_{i+h} + w'_{i+h} \qquad (5.68)$$

where

$$w'_{i+h} = \frac{1}{k_p^2} \sum_{j=0}^{p} w_{i+h+j}y^*_j \qquad (5.69)$$

$$J_{i+h} = \sum_{j=1}^{p} s_{i+h+j}a'_j \qquad (5.70)$$

and $$a'_k = \frac{1}{k_p^2} \sum_{j=0}^{p-k} y_j y^*_{j+k} \qquad (5.71)$$

for any possible value of k. It can be shown[A9] that, the real and imaginary parts of the filtered noise components $\{w'_j\}$ in eqns. 5.68 and 5.69 are Gaussian random variables with zero mean and variance σ^2/k_p^2. Thus, the value of x'_{i+h} in eqn. 5.68 consists of three components, namely the signal component s_{i+h} (whose value is to be determined here

as the value of x_{i+h}), the intersymbol interference component J_{i+h}, and the noise component w'_{i+h}. Consequently, the value of x'_{i+h} here is treated as the estimated value of the 16-point QAM data symbol s_{i+h}, and the selection process now to be described uses this value of x'_{i+h} to select, from the 16 possible values of s_{i+h}, a total of 1, 2, 4, or 9 values for the component x_{i+h}. Four selection processes having the four values of m'_h (that is, 1, 2, 4, and 9) have been considered here. It should be noticed that when $m'_h = 16$, all the 16 possible values of s_{i+h} are used for the component x_{i+h}, and hence no selection process is required here. In order to simplify the nomenclature for the selection processes here, rename s_{i+h} as s_h, x_{i+h} as x_h, x'_{i+h} as x_h, J_{i+h} as J_h, and w'_{i+h} as w'_h. Eqn. 5.68 is now rewritten as

$$x'_h = s_h + J_h + w'_h \qquad (5.72)$$

or

$$\begin{aligned} \mathrm{Re}(x'_h) &= \mathrm{Re}(s_h) + \mathrm{Re}(J_h) + \mathrm{Re}(w'_h) \\ \mathrm{Im}(x'_h) &= \mathrm{Im}(s_h) + \mathrm{Im}(J_h) + \mathrm{Im}(w'_h) \end{aligned} \qquad (5.73)$$

where Re(.) and Im(.) are the real and imaginary parts of (.), and the selection process now uses the estimate x'_h (of s_h) to select m'_h possible values of s_h for the component x_h here.

Selection process A ($m'_h = 1$)

This selection process operates on the estimate x'_h (eqn. 5.73), of s_h, to select the possible value of x_h that is closest to x'_h, where x_h may have any of the 16 possible complex values of s_h given by $a_h + jb_h$ for $a_h = \pm 1, \pm 3$ and $b_h = \pm 1, \pm 3$. The selection process here is carried out by comparing the real and imaginary parts of x'_h with the threshold

values of 2, 0, and -2, as is shown in Table 5.6.

Thus, effectively, the selection process here operates by partitioning the 16-point QAM signal constellation of s_h into the 16 regions shown in Fig. 5.14, and it takes the value of x_h to be the possible value of s_h that lies in the same region as that of the value of x_h', bearing in mind that x_h' is here treated as the estimate of s_h.

$\mathrm{Re}(x_h)$ or $\mathrm{Im}(x_h)$	=	-3	if	$-2 \geq \mathrm{Re}(x_h')$ or $\mathrm{Im}(x_h')$
$\mathrm{Re}(x_h)$ or $\mathrm{Im}(x_h)$	=	-1	if	$0 \geq \mathrm{Re}(x_h')$ or $\mathrm{Im}(x_h') > -2$
$\mathrm{Re}(x_h)$ or $\mathrm{Im}(x_h)$	=	1	if	$2 \geq \mathrm{Re}(x_h')$ or $\mathrm{Im}(x_h') > 0$
$\mathrm{Re}(x_h)$ or $\mathrm{Im}(x_h)$	=	3	if	$\mathrm{Re}(x_h')$ or $\mathrm{Im}(x_h') > 2$

Table 5.6 Selection of the possible value of x_h closest to the estimated value x_h' (eqn. 5.73) of s_h. Re(.) and Im(.) are the real and imaginary parts of (.).

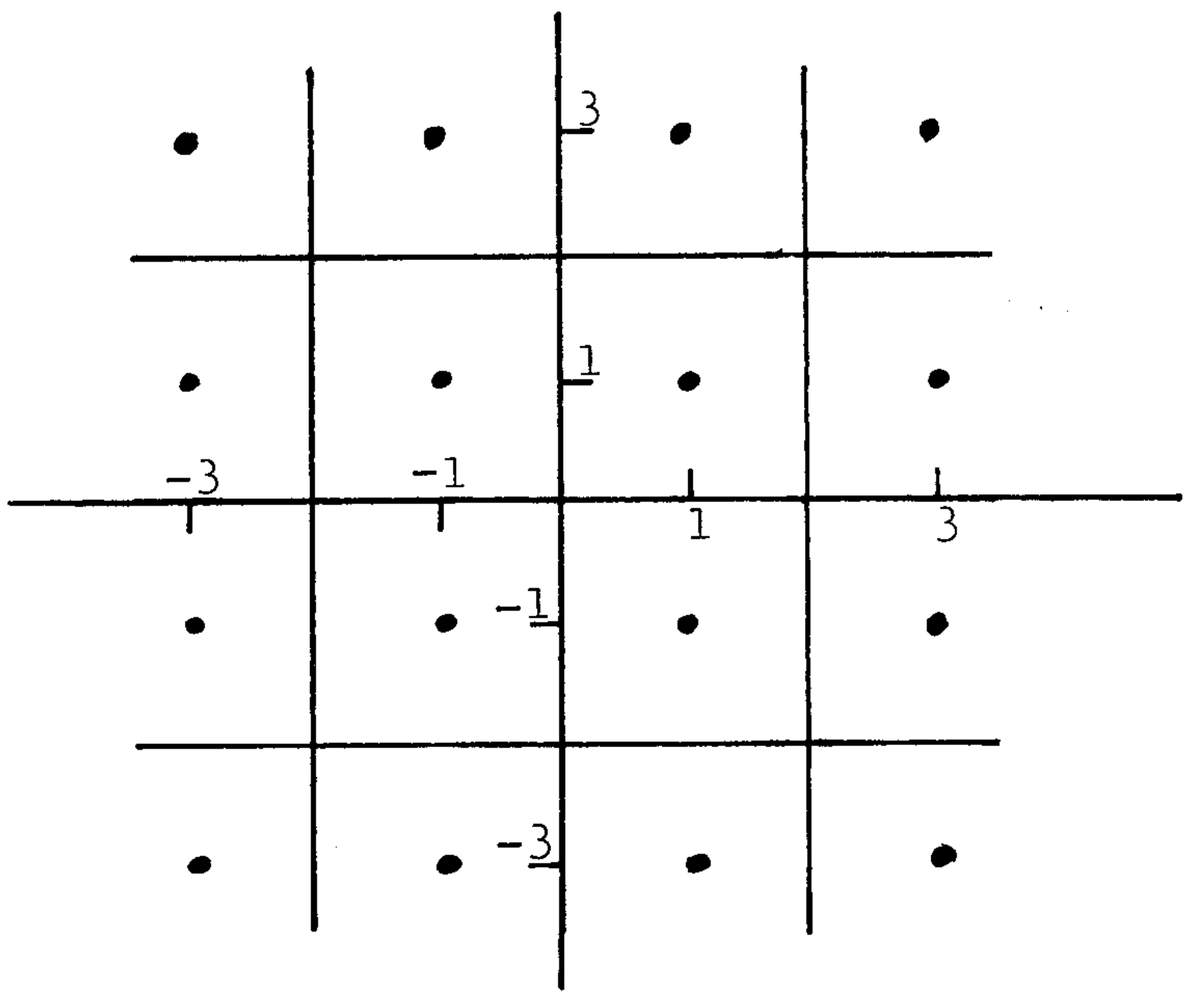


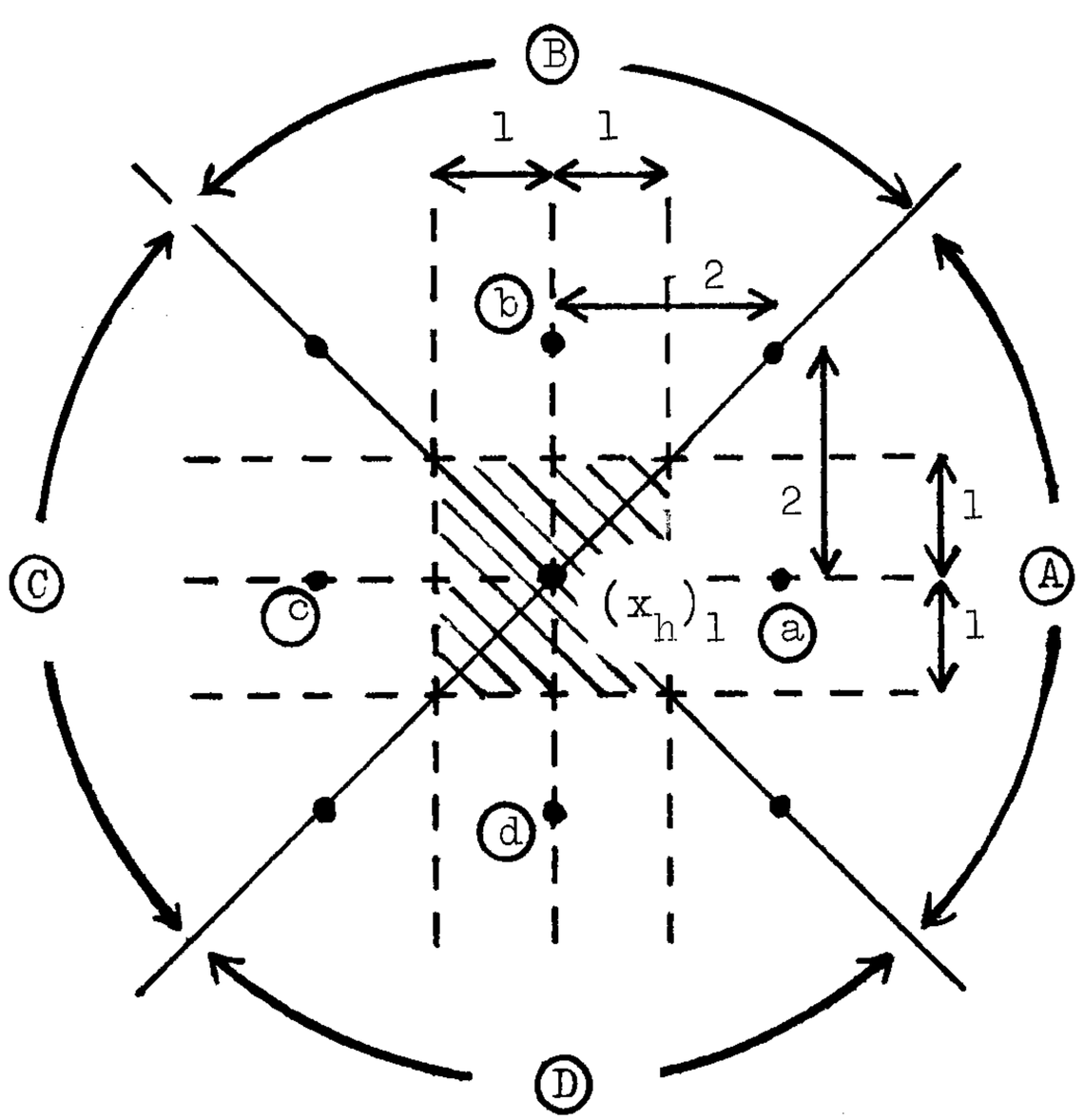
Fig. 5.14 A 16-point QAM signal constellation of s_h partitionned into 16 regions having the 16 possible values (represented as points here) of s_h.

Selection process B ($m'_h = 2$)

This selection process operates on the estimate x'_h (eqn. 5.73), of s_h, to select the 2 possible values of x_h that are closest to the value of x'_h, where x_h may have any of the 16 possible complex values of s_h.

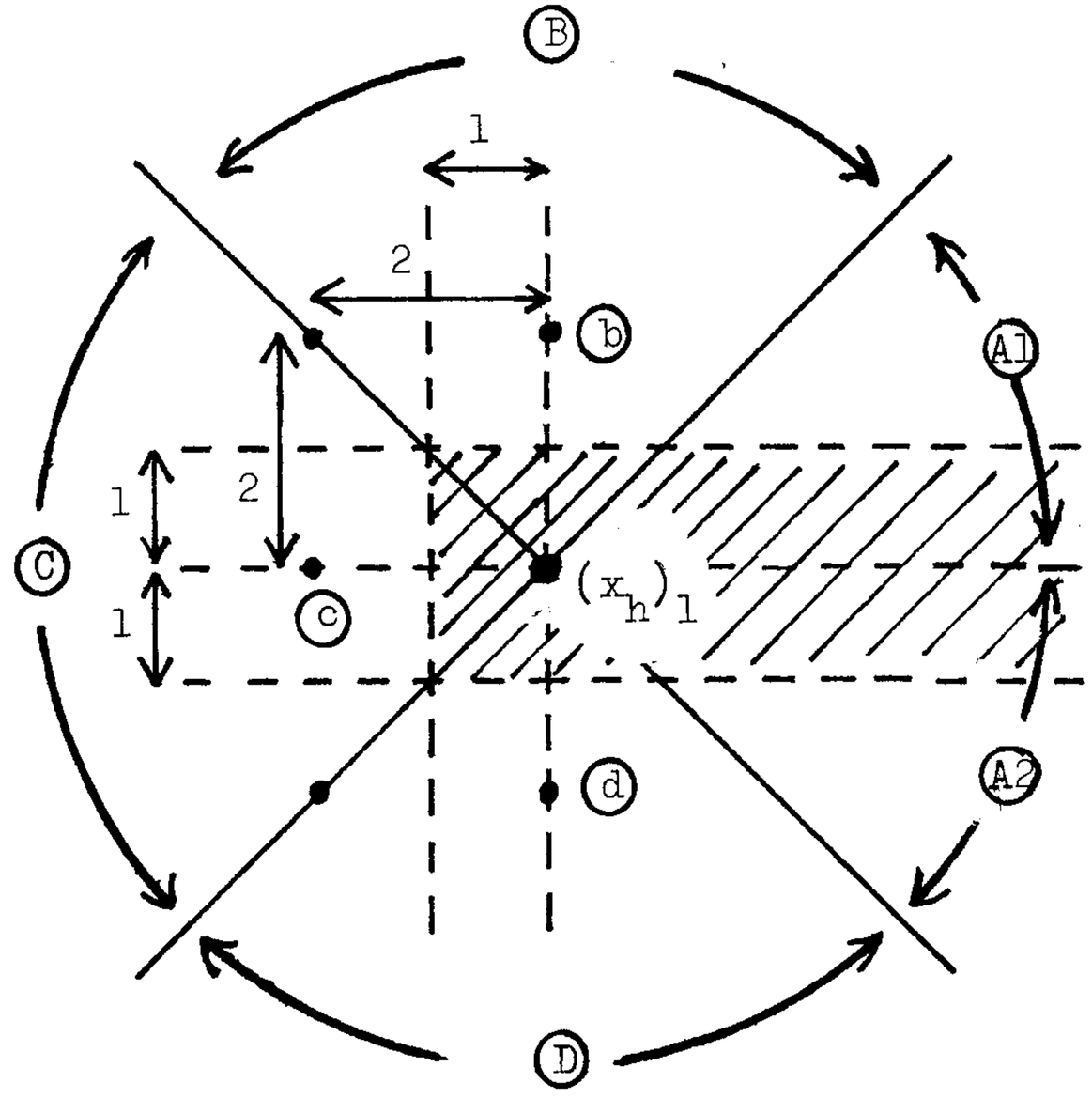
Thus, the possible value of x_h that is cloest to the value of x'_h is first selected here by using the method outlined in Table 5.6. Let this value of x_h be designated as $(x_h)_1$. The selection process then proceeds to select the second possible value of x_h that is cloest to the value of x'_h according to the principle described below. Let this second selected value of x_h be designated as $(x_h)_2$.

Being a possible value of the 16-point QAM signal s_h (Fig. 5.14), the first selected value $(x_h)_1$ of x_h here may be one of the 4 central points ($\pm 1 \pm j$), or one of the 4 cornor points ($\pm 3 \pm 3j$), or one of the 8 side points ($\pm 1 \pm 3j$ or $\pm 3 \pm j$), in the 16-point QAM signal constellation. Fig. 5.15 shows a possible arrangement for each of the three cases just mentioned. The estimate x'_h (of s_h) is here restricted to lie within the shaded area that contains the first selected value $(x_h)_1$ of x_h, as can be seen from Table 5.6 and Fig. 5.14, bearing in mind $(x_h)_1$ is selected according to Table 5.6. Consider first the arrangement of Fig. 5.15a where $(x_h)_1$ is one of the 4 central points in the 16-point QAM signal constellation. It can be seen here that, so long as x'_h lies in the shaded area of region (A), then (a) is the possible value of x_h that is cloest to x'_h (other than the central point $(x_h)_1$) and should therefore be taken as the second selected value $(x_h)_2$ here. Similarly, if x'_h lies in region (B), (C), or (D), then (b), (c), or (d) respectively should be taken as the value of $(x_h)_2$. Consider next the arrangement of Fig. 5.15b where the point (a) in Fig. 5.15a is absent here. Clearly,



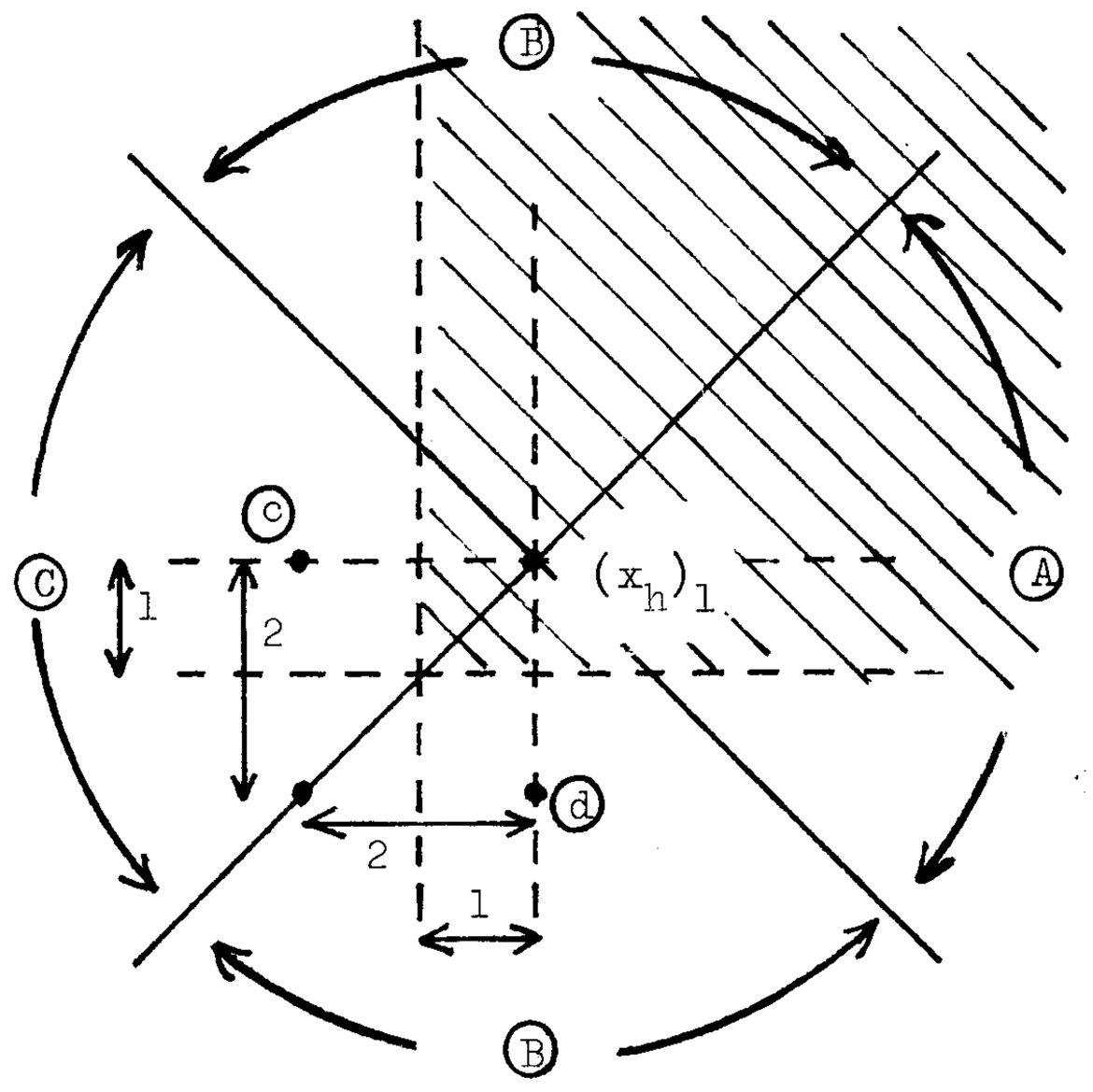
(a)

When $(x_h)_1$ is one of the 4 central points.



(b)

When $(x_h)_1$ is one of the 8 side points.



(c)

When $(x_h)_1$ is one of the 4 corner points.

Fig. 5.15 The first selected value $(x_h)_1$ (Table 5.6) and its neighbouring points in the 16-point QAM signal constellation. The shaded area is where the estimate x'_h (of s_h) lie.

if x_h' lies in any of the regions (B), (C), and (D), then the selection process used in Fig. 5.15a should be used here to select the value of $(x_h)_2$. If, however, x_h' lies in region (A_1), then it can be seen from Fig. 5.15b that, (b) is now the possible value of x_h that is closest to x_h' and should therefore be taken as the value of $(x_h)_2$. Similarly, if x_h' lies in region (A_2), then (c) becomes the point closest to x_h' and should therefore be taken as $(x_h)_2$. Finally, consider the arrangement shown in Fig. 5.15c where both the points (a) and (b) in Fig. 5.15a are absent here. Again, so long as x_h' lies in any of the regions (C) and (D), then the selection process used in Fig. 5.15a should be used here to select the value of $(x_h)_2$. If, however, x_h' lies in region (A), then it can be seen from Fig. 5.15c that (d) is now closest to x_h' and should therefore be taken as the value of $(x_h)_2$. Similarly, if x_h' lies in region (B), then (c) should be taken as the value of $(x_h)_2$.

Thus, the principle outlined above is used for the selection process here to select the second possible value $(x_h)_2$ of x_h that is closest to the estimate x_h' of s_h, bearing in mind that the first possible value $(x_h)_1$ of x_h that is closest to x_h' has already been selected using Table 5.6. The selection process now operates by first evaluating the two quantities

$$\Delta_1 = \mathrm{Re}(x_h') - \mathrm{Re}((x_h)_1)$$

$$\text{and} \qquad \Delta_2 = \mathrm{Im}(x_h') - \mathrm{Im}((x_h)_1) \qquad (5.74)$$

where Re(.) and Im(.) are the real and imaginary parts of (.) respectively. In order to locate the region where x_h' lies (Fig. 5.15), the absolute value of Δ_1 is compared with that of Δ_2. Two cases may now occur depending on whether $|\Delta_1|$ is larger or smaller than $|\Delta_2|$, where $|(.)|$ is the

absolute value of (.). Consider first when $|\Delta_1|$ is larger than $|\Delta_2|$, which implies that the estimate x_h' must now lie in any of the regions (A) and (C) in Fig. 5.15. Thus, the second selected value $(x_h)_2$ here is determined as

$$\text{Re}((x_h)_2) = \text{Re}((x_h)_1) + 2\ \text{sgn}(\Delta_1)$$

$$\text{and} \qquad \text{Im}((x_h)_2) = \text{Im}((x_h)_1) \qquad (5.75)$$

where sgn(.) is the sign of (.) and is either 1 or -1. The value of $\text{Re}((x_h)_2)$ just determined is then checked and if it is not one of the 4 possible values (± 1, ± 3) of $\text{Re}(x_h)$ (as may occur in Fig. 5.15b or 5.15c), then the value of $(x_h)_2$ is changed to

$$\text{Re}((x_h)_2) = \text{Re}((x_h)_1)$$

$$\text{and} \qquad \text{Im}((x_h)_2) = \text{Im}((x_h)_1) + 2\ \text{sgn}(\Delta_2) \qquad (5.76)$$

The value of $\text{Im}((x_h)_2)$ just determined is again checked and if it is not one of the 4 possible values (± 1, ± 3) of $\text{Im}(x_h)$ (as may occur in Fig. 5.15c), then the value of $(x_h)_2$ is finally changed to

$$\text{Re}((x_h)_2) = \text{Re}((x_h)_1)$$

$$\text{and} \qquad \text{Im}((x_h)_2) = \text{Im}((x_h)_1) - 2\ \text{sgn}(\Delta_2) \qquad (5.77)$$

Consider now the case when $|\Delta_1|$ is smaller than or equal to $|\Delta_2|$, which implies that the estimate x_h' must now lie in any of the regions (B) and (D) in Fig. 5.15. Thus, the second selected value $(x_h)_2$ here is taken to be that given by eqn. 5.76. This value is then checked and if it is not a possible value of x_h, then the value of $(x_h)_2$ is

changed to that given by eqn. 5.75. The new value of $(x_h)_2$ is again checked and if it is not a possible value of x_h, then the value of $(x_h)_2$ is finally changed to

$$\text{Re}((x_h)_2) = \text{Re}((x_h)_1) - 2\,\text{sgn}(\Delta_1)$$

$$\text{and} \qquad \text{Im}((x_h)_2) = \text{Im}((x_h)_1) \qquad (5.78)$$

The selection process just described is summarised in Table 5.7.

	When $\lvert\Delta_1\rvert > \lvert\Delta_2\rvert$	When $\lvert\Delta_1\rvert \leq \lvert\Delta_2\rvert$
1st solution of $(x_h)_2$	eqn. 5.75	eqn. 5.76
2nd solution of $(x_h)_2$ *	eqn. 5.76	eqn. 5.75
3rd solution of $(x_h)_2$ **	eqn. 5.77	eqn. 5.78

Table 5.7 Selection of the second possible value $(x_h)_2$ of x_h closest to the estimated value x_h' (eqn. 5.73) of s_h.

* means 'evaluated only if 1st solution is not a possible value of x_h'.

** means 'evaluated only if both 1st and 2nd solutions are not possible values of x_h'.

$|\Delta_1|$ and $|\Delta_2|$ are the absolute values of Δ_1 and Δ_2 (eqn. 5.74) respectively.

Selection process C $(m_h' = 4)$

This selection process operates on the estimate x_h' (eqn. 5.73), of s_h, to select 4 possible values of x_h, where x_h may have any of the 16 possible complex values of s_h. The selection process here is such that, the 2 possible values of $Re(x_h)$ that are closest to the value of $Re(x_h')$ and the 2 possible values of $Im(x_h)$ that are closest to the value of $Im(x_h')$ are selected to give a total of 4 possible values of x_h, where Re(.) and Im(.) are, of course, the real and imaginary parts of (.) respectively. Thus, this selection process is carried out by comparing (separately) the real and imaginary parts of x_h' with the 2 threshold values 1 and -1, as is shown in Fig. 5.16 and Table 5.8.

$Re(x_h)$ or $Im(x_h)$ = -3 and -1	if	$-1 \geq Re(x_h')$ or $Im(x_h')$
$Re(x_h)$ or $Im(x_h)$ = -1 and 1	if	$1 \geq Re(x_h')$ or $Im(x_h') > -1$
$Re(x_h)$ or $Im(x_h)$ = 1 and 3	if	$Re(x_h')$ or $Im(x_h') > 1$

Table 5.8 Selection of the 2 possible values of $Re(x_h)$ (or $Im(x_h)$) that are closest to the value of $Re(x_h')$ (or $Im(x_h')$) given by eqn. 5.73. Re(.) and Im(.) are the real and imaginary parts of (.) respectively.

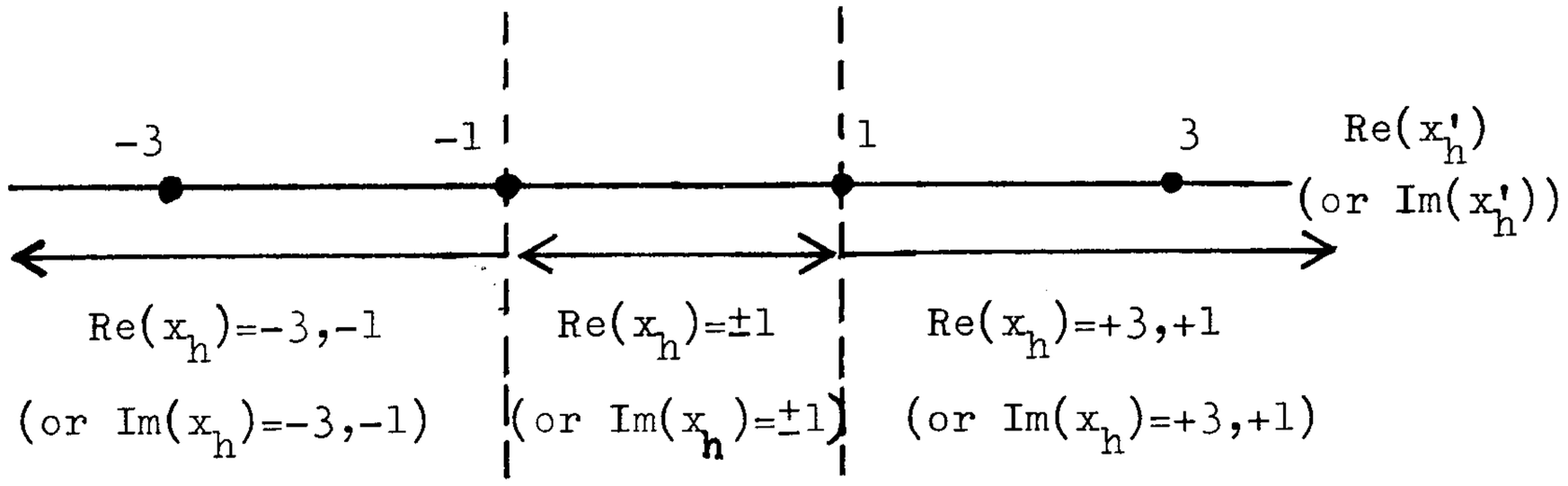


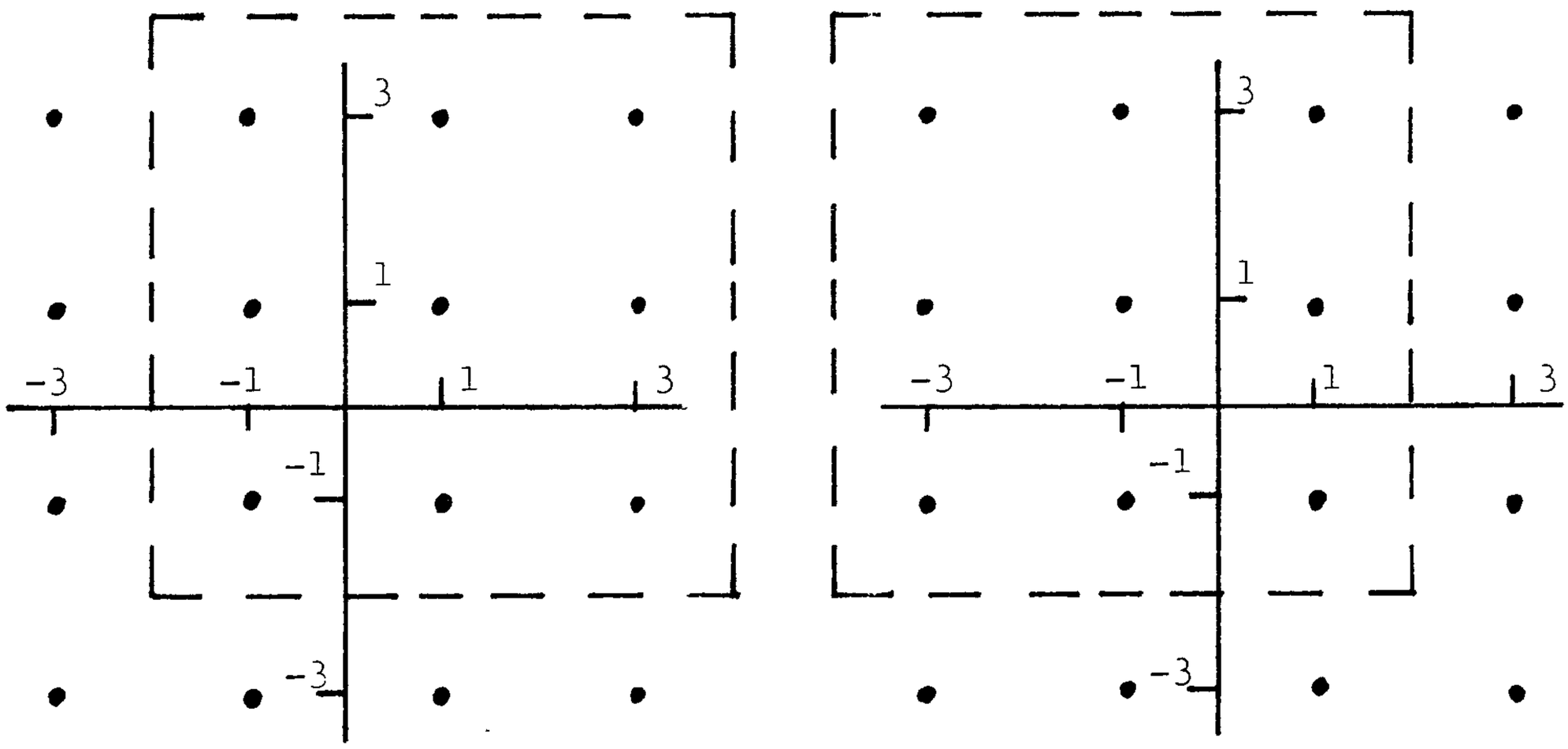
Fig. 5.16 Selection of the 2 possible values of $Re(x_h)$ (or $Im(x_h)$) that are closest to the value of $Re(x_h')$ (or $Im(x_h')$).

Selection process D ($m_h' = 9$)

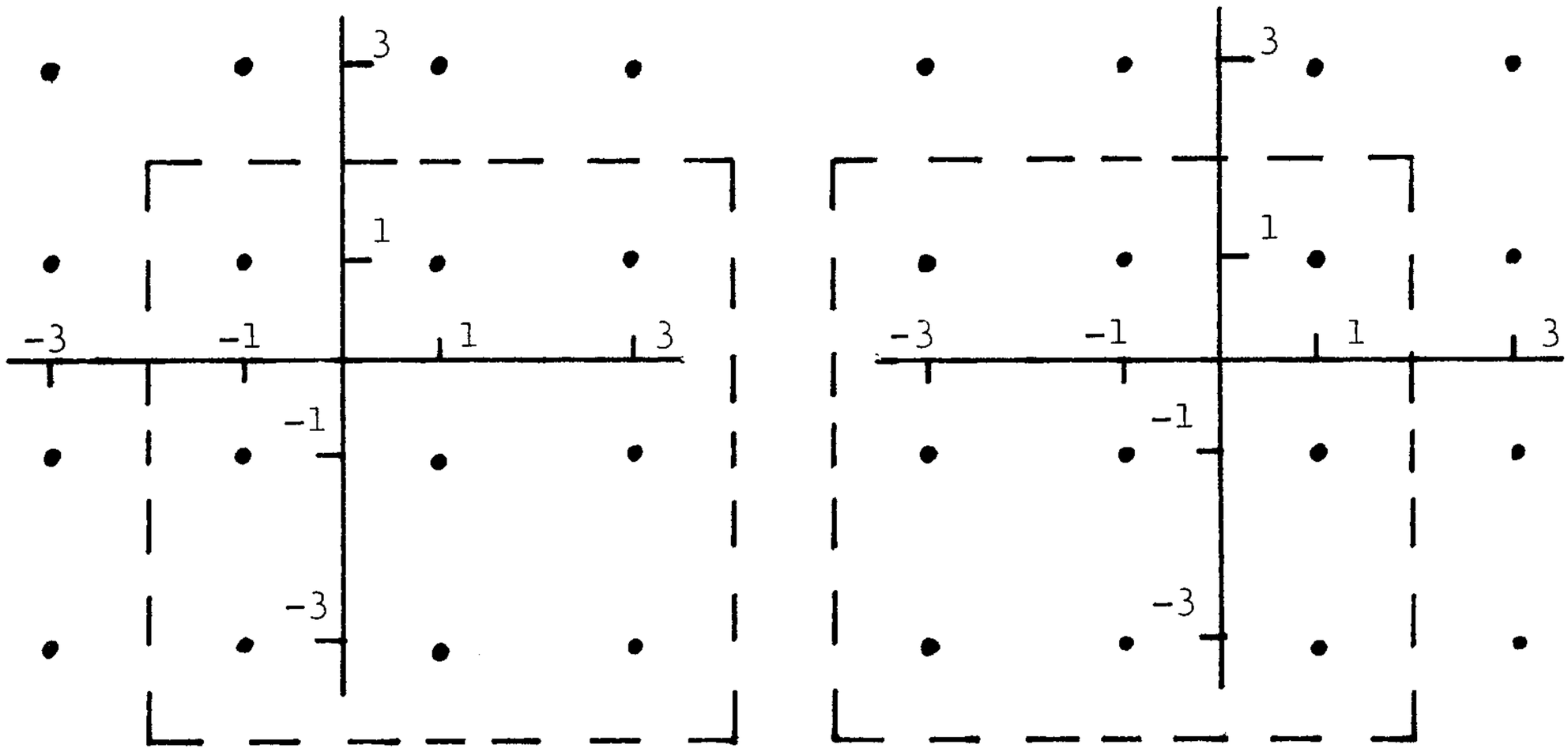
This selection process operates on the estimate x_h' (eqn. 5.73) to select 9 possible values of x_h, where x_h may have any of the 16 possible complex values of s_h (eqn. 5.2). The selection process here is such that the 3 possible values of $Re(x_h)$ that are closest to $Re(x_h')$, and the 3 possible values of $Im(x_h)$ that are closest to $Im(x_h')$ are selected to give a total of 9 possible values of x_h, where Re(.) and Im(.) are the real and imaginary parts of (.) respectively. This selection process is carried out by comparing (separately) the real and imaginary parts of x_h' with the threshold value of 0, as is shown in Fig. 5.17 and Table 5.9.

$Re(x_h)$ or $Im(x_h)$ = -3 and -1 and 1	if	$Re(x_h')$ or $Im(x_h') \leq 0$
$Re(x_h)$ or $Im(x_h)$ = -1 and 1 and 3	if	$Re(x_h')$ or $Im(x_h') > 0$

Table 5.9 Selection of the 3 possible values of $Re(x_h)$ (or $Im(x_h)$) that are closest to the value of $Re(x_h')$ (or $Im(x_h')$) given by eqn. 5.73. Re(.) and Im(.) are the real and imaginary parts of (.) respectively.



(a) (c)



(b) (d)

Fig. 5.17 Selection of the 9 possible values of x_h (Table 5.9).

(a) when $Re(x_h') > 0$ and $Im(x_h') > 0$.

(b) when $Re(x_h') > 0$ and $Im(x_h') \leq 0$.

(c) when $Re(x_h') \leq 0$ and $Im(x_h') > 0$.

(d) when $Re(x_h') \leq 0$ and $Im(x_h') \leq 0$.

Thus, any of the 4 selection processes just described can be used to derive m'_h possible values of s_{i+h} (eqn. 5.68) for the component x_{i+h} $(0 \leq h \leq \lambda)$ in the expanded vectors $\{P'_i\}$ (eqn. 5.54 and Fig. 5.12), in the detection process of system 7, where m'_h may have any of the four values 1, 2, 4, and 9. It is desirable that the transmitted value of the data symbol s_{i+h} be included as one of the m'_h selected values of x_{i+h}, so that the correct detection of s_{i+h} can be achieved here. The necessary condition for the value of s_{i+h} to be included as one of the m'_h selected values of x_{i+h} have been derived in appendix A8 for each of the four selection processes here, and the results are shown in Table 5.10. Furthermore, since x'_{i+h} (eqn. 5.68) is here treated as the estimate of s_{i+h}, it can be seen from eqn. 5.68 that the value of s_{i+h} is more likely to be included as one of the m'_h selected values of x_{i+h} here if the magnitude of the intersymbol interference component J_{i+h} and the variance of the noise components $\{w'_{i+h}\}$ in eqn. 5.68 are smaller. The value of J_{i+h} is given by eqn. 5.70, and its magnitude can be seen to be such that

$$|J_{i+h}| \leq \sum_{k=1}^{p} |s_{i+h-k}||a'_k| \leq 3\sqrt{2} \sum_{k=1}^{p} |a'_k| \qquad (5.79)$$

where the $\{a'_k\}$ here are as given by eqn. 5.71, and p+1 is the number of taps of the transversal filter (Fig. 5.13) used in the detector of system 7. The real and imaginary parts of the noise components $\{w'_{i+h}\}$ in eqn. 5.68 are Gaussian random variables with zero mean and variance σ^2/k_p^2, where $2\sigma^2$ is the two-sided power spectral density of the noise added at the output of the transmission path shown in Fig. 2.3, and k_p^2 is as given by eqn. 5.65. It therefore follows that, the number of taps, p+1, to be used in the transversal filter (Fig. 5.13) of system 7 should be selected to be such that, a small value of $|J_{i+h}|$

or $3\sqrt{2}\sum_{k=1}^{p}|a'_k|$ (eqn. 5.79), and a large value of k_p (that is, a small value of σ^2/k_p^2) are achieved here. In the absence of noise ($w'_{i+h}= 0$), Table 5.10 suggests that the value of s_{i+h} is always included as one of the m'_h selected values of x_{i+h} here so long as $|J_{i+h}| < v_h$. It therefore appears that at high signal to noise ratios, the value of p to be used here should be further restricted to be such that, the value of $3\sqrt{2}\sum_{k=1}^{p}|a'_k|$ (eqn. 5.79) does not largely exceed that of v_h. The values of k_p (eqn. 5.65) and $3\sqrt{2}\sum_{k=1}^{p}|a'_k|$ (eqn. 5.71) have been calculated for system 7 operating over channel A (Table 2.4) at various values of p, and the results are shown in Table 5.11. As Table 5.11 shows, system 7 with p = 3 is very likely to have a very poor performance because its associated value of $3\sqrt{2}\sum_{k=1}^{p}|a'_k|$ largely exceeds any of the 4 possible values of v_h given in Table 5.10. Consequently, only the values of 1 and 2 for p are considered in the evaluation of the system performance here.

Computer simulation tests have been carried out to determine the tolerance to Gaussian noise of system 7 operating over channel A (Table 2.4), and the results are shown in Figs. 5.18 - 5.21. The 95% confidence limits of these results are about $\pm$0.5 dB. The bit error rate and the signal to noise ratio here are as defined by eqns. 5.3 and 5.4 respectively. The number m' of vectors $\{P'_i\}$ (eqn. 5.54) expanded from each of the m stored vectors $\{X_{i-1}\}$ (eqn. 5.8) here are restricted to be about 16 so that the total number of measurements of $\{C'_i\}$ (eqn. 5.56) per data symbol involved here is about the same as the total number of cost measurements (eqns. 5.11 - 5.13) per data symbol involved in system 1. This is to ensure that, the total number of operations per data symbol involved in the detection process of system 7 does not greatly exceeds that involved in the detection process of system 1, bearing in mind that system 1 uses the arrangement where an 'all-pass' transversal filter is inserted ahead of a near-maximum likelihood detector.

Number of selected values for x_{i+h} m'_h	Condition for the transmitted value of s_{i+h} to be included as one of the m'_h selected values for x_{i+h} (appendix A8)
1	$\left\|J_{i+h} + w'_{i+h}\right\| < v_h = 1$
2	$\left\|J_{i+h} + w'_{i+h}\right\| < v_h = \sqrt{2}$
4	$\left\|J_{i+h} + w'_{i+h}\right\| < v_h = 2$
9	$\left\|J_{i+h} + w'_{i+h}\right\| < v_h = 3$

Table 5.10 The necessary conditions for the transmitted value of s_{i+h} (eqn. 5.68) to be included as one of the selected values for the component x_{i+h} $(0 \leq h \leq \lambda)$ in the expanded vectors $\{P'_i\}$ (eqn. 5.54) in the detection process of system 7. J_{i+h} and w'_{i+h} are the intersymbol interference component and noise component respectively (eqn. 5.68).

p	k_p	σ^2/k_p^2	$3\sqrt{2}\sum_{k=1}^{p}\left\|a'_k\right\|$
1	0.188	28.2936^2	0.564
2	0.527	3.6016^2	1.658
3	0.751	1.7736^2	3.593

Table 5.11 The maximum intersymbol interference level $3\sqrt{2}\sum_{k=1}^{p}\left|a'_k\right|$ (eqn. 5.79) and the noise variance σ^2/k_p^2 in the estimate x'_{i+h} of s_{i+h} in eqn. 5.68. The $\{a'_k\}$ and k_p are as given by eqns. 5.71 and 5.65 respectively, and p+1 is the number of taps used in the filter (Fig. 5.13) of system 7.

It can be seen from Fig. 5.18 that, system 7 with p = 1 has a very poor tolerance to noise, where p+1 is the number of taps of the filter (Fig. 5.13) used in deriving the values for the components $x_i, x_{i+1}, \cdots, x_{i+\lambda}$ in the expanded vectors $\{P_i'\}$ (eqn. 5.54) here. One possible reason to this poor performance is that, the noise variance σ^2/k_p^2 (Table 5.11) involved in deriving the values for the component x_{i+h} $(0 \leq h \leq \lambda)$ is very high here so that the transmitted value of the data symbol s_{i+h} is very often not included as one of these derived values for x_{i+h}, leading to a high probability of error in the detection of s_{i+h}. Figs. 5.18 - 5.21 also show that at high signal to noise ratios, system 7 with a small value (e.g. 4) of m_0' (number of possible values of s_i selected for the component x_i in the vectors $\{P_i'\}$) appears to have a poor tolerance to noise. This is probably because the intersymbol interference level (eqns. 5.68 and 5.79 and Table 5.11) involved in deriving the values for x_i here is so high that a large number of values for x_i must now be selected so as to include the transmitted value of s_i . Furthermore, if x_i does not have the correct value of s_i here, then the intersymbol interferences of s_i involved in deriving the values for the components $x_{i+1}, x_{i+2}, \cdots, x_{i+\lambda}$ are increased instead of being removed, so that the transmitted values of $s_{i+1}, s_{i+2}, \cdots, s_{i+\lambda}$ are now less likely to be included in the corresponding selected values for $x_{i+1}, x_{i+2}, \cdots, x_{i+\lambda}$. As Figs. 5.18 - 5.21 show, system 7 with the various arrangements tested here appear to have the best performance when $m_0' = 9$, $m_1' = 2$, and $m_2' = 1$ (Fig. 5.20), where m_h' is the number of selected values for the component x_{i+h} $(0 \leq h \leq \lambda)$ in the vectors $\{P_i'\}$ here (eqn. 5.54 and Fig. 5.12). However, this performance is only slightly better than that of the optimum nonlinear equalizer when operating over channel A here, as can be seen from Fig. 5.20. System 7 therefore does not appear to be promising when operating over channel A here, since

it requires many more operations per data symbol than that required in the optimum nonlinear equalizer while having only a slight improvement in tolerance to noise over the equalizer here.

One weakness of system 7 appears to be that, the intersymbol interference levels involved in deriving the values for the components $x_i, x_{i+1}, \cdots, x_{i+\lambda}$ in the expanded vectors $\{P_i'\}$ (eqn. 5.54 and Fig. 5.12) here are often so high that a large number of values for $x_i, x_{i+1}, \cdots, x_{i+\lambda}$ are now required to be selected and used in the detection process of system 7 so as to achieve a satisfactory tolerance to noise. Further modifications to system 7 should therefore aim at reducing these intersymbol interference levels.

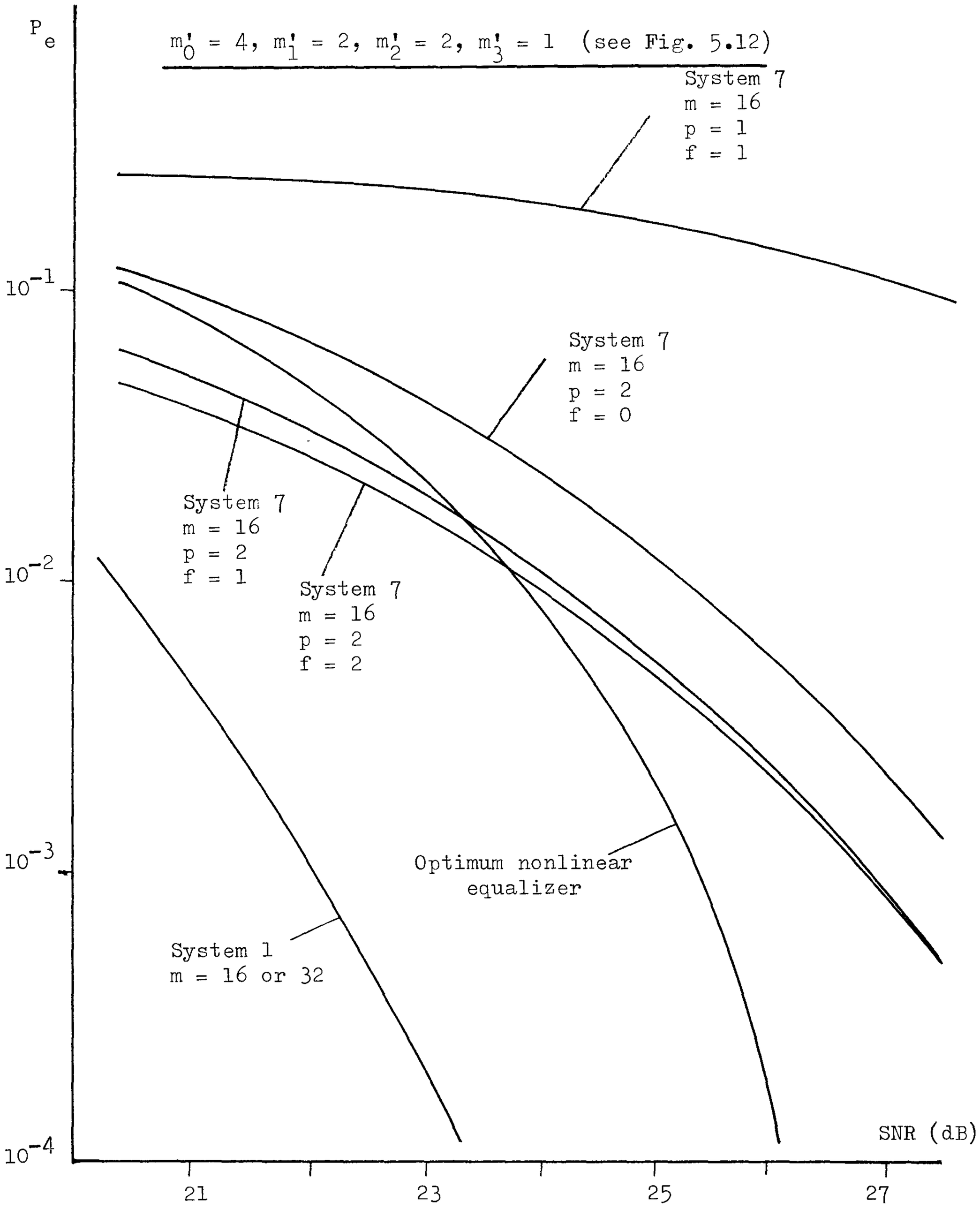


Fig. 5.18 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 7 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16. m is the number of stored vectors, p+1 is the number of taps of the filter of Fig. 5.13, and y_f is the first significant component of the channel sampled impulse response.

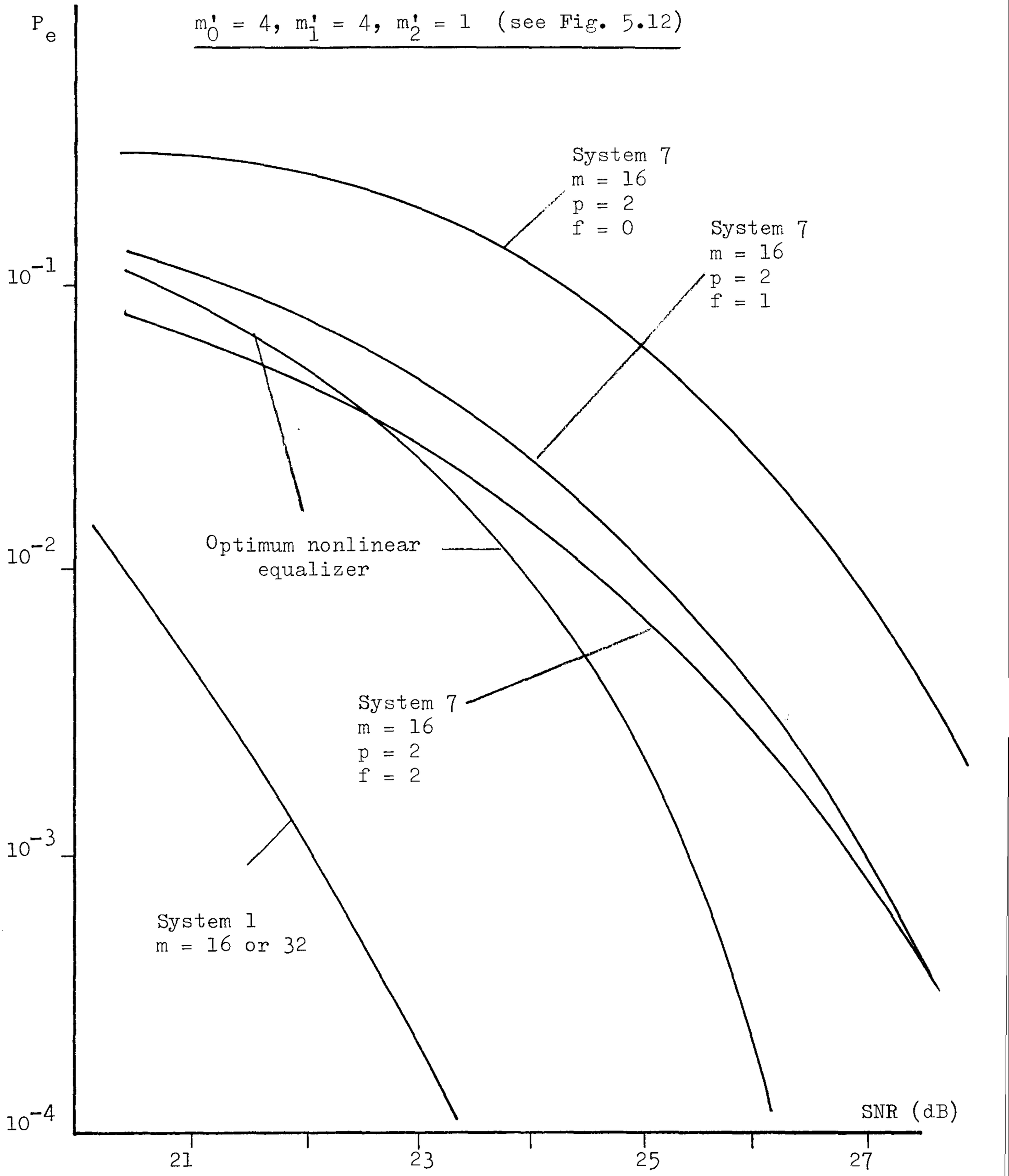


Fig. 5.19 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 7 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16. m, p, and f are as in Fig. 5.18.

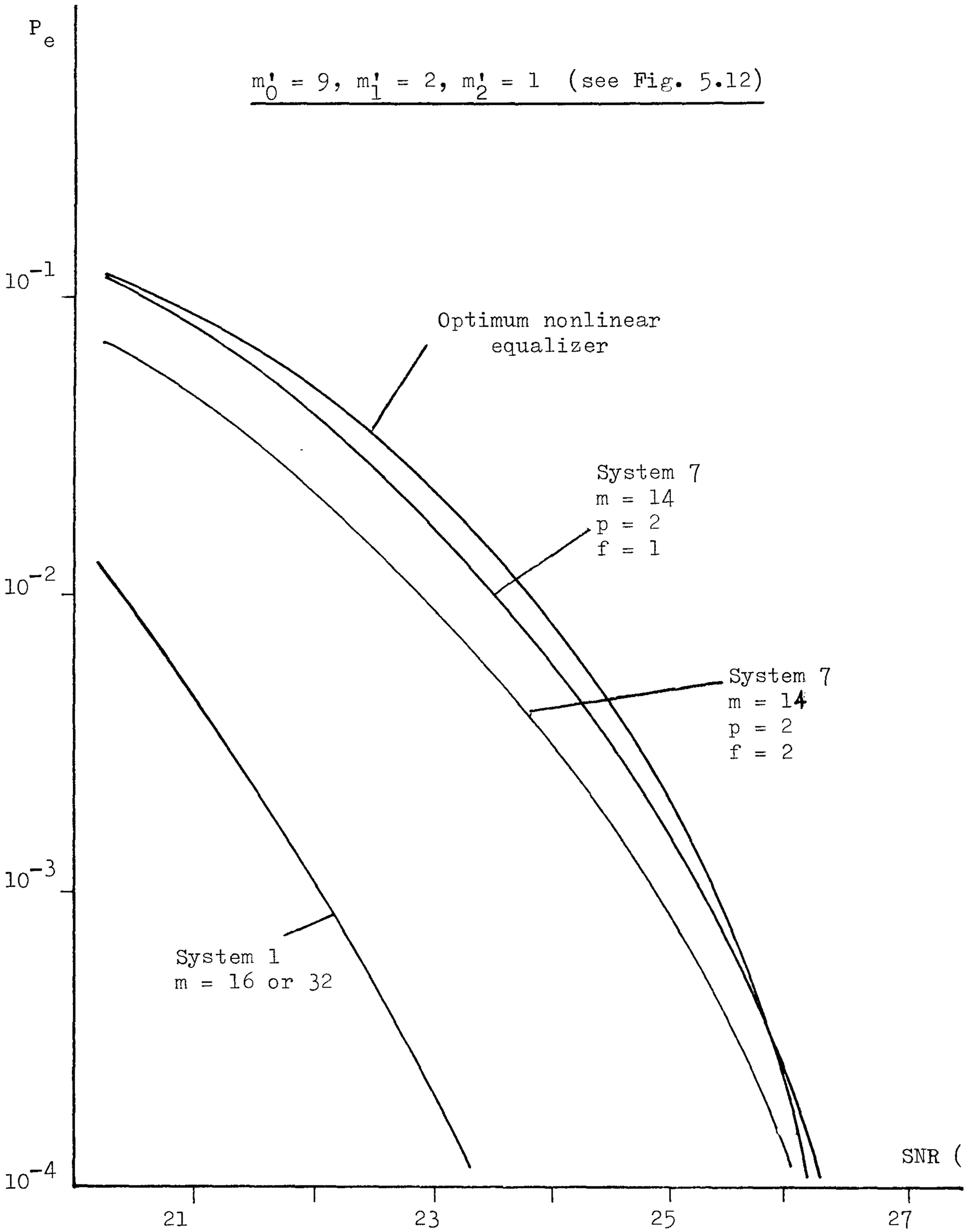


Fig. 5.20 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 7 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16. m, p, and f are as in Fig. 5.18.

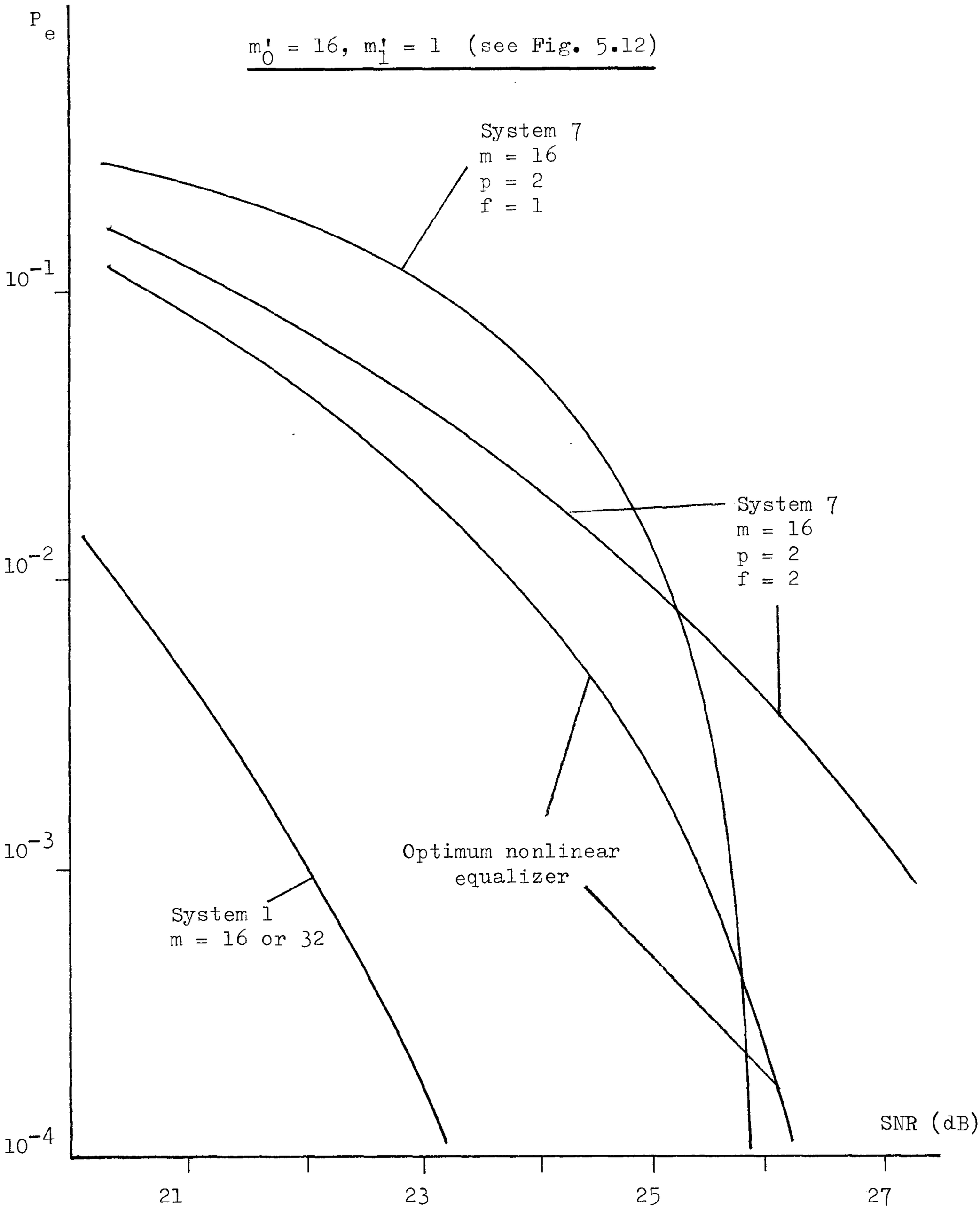


Fig. 5.21 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 7 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16. m, p, and f are as in Fig. 5.18.

5.10 System 8

In the detection process of system 7, each of the m n-component stored vectors $\{X_{i-1}\}$ (eqn. 5.8) is expanded into m' $(n+\lambda+1)$-component vectors $\{P'_i\}$ (eqn. 5.54 and Fig. 5.12) having the m' possible values for their last $\lambda+1$ components $x_i, x_{i+1}, \cdots, x_{i+\lambda}$. These $\lambda+1$ components are derived, one at a time using the linear filter shown in Fig. 5.13. System 8 is a modification of system 7 in that two linear filters (instead of one in system 7) are now used to derive the values, for two components at a time, for the $\lambda+1$ components $x_i, x_{i+1}, \cdots, x_{i+\lambda}$ here. The idea behind this modification is that, the intersymbol interference of, say s_{j+1}, in system 7 in deriving the values for x_j is now 'eliminated' by regarding s_{j+1} also as a signal component (to be determined as x_{j+1}) here, bearing in mind that the data symbol s_j is a signal component (to be determined as x_j) in the derivation process for x_j and x_{j+1} here. The operation of system 8 is otherwise similar to that of system 7.

Thus, the basic model of the signal processor for system 8 is the same as that shown in Fig. 5.9. The intersymbol interference canceller in Fig. 5.9 operates on the received sample values $r_i, r_{i+1}, \cdots, r_{i+\alpha}$ (eqn. 5.1) to give, at the detector input, the sample values $r'_{i,0}, r'_{i,1}, \cdots, r'_{i,\alpha}$ (eqn. 5.52), as is described before for system 6 or 7. The near-maximum likelihood detector (Fig. 5.9) operates on its input sample values (eqn. 5.52) to give at its output the detected data symbol s'_{i-n}, and this is described as below.

Just prior to the receipt of the samples $\{r'_{i,h}\}$ (eqn. 5.52), the detector of system 8 holds in store m n-component vectors $\{X_{i-1}\}$ (eqn. 5.8) together with the associated costs $\{C_{i-1}\}$ (eqns. 5.10, 5.11, and 5.16). On receiving $\{r'_{i,h}\}$, each of the m stored vectors $\{X_{i-1}\}$ is expanded

into m' (n+1)-component vectors $\{P_i\}$ (eqn. 5.9), where m' may have any of the values 4, 9, and 16 here. The first n components of these m' vectors $\{P_i\}$ are as in the original vector X_{i-1}, and the last component x_i may have any m' (4, 9, or 16) of the 16 possible values of the data symbol s_i. The derivation of the m' values of x_i here will be described in details later. The detector now evaluates the cost C_i for each of the (m)(m') expanded vectors $\{P_i\}$ by using eqns. 5.11 and 5.16. Each of these (m)(m') vectors $\{P_i\}$ is then further expanded into a (n+3)-component vector P_i', where

$$P_i' = \begin{bmatrix} x_{i-n} & x_{i-n+1} & \cdots & x_i & x_{i+1} & x_{i+2} \end{bmatrix} \qquad (5.80)$$

which is the same vector defined by eqn. 5.54 so long as the value of λ in eqn. 5.54 is equal to 2. The first n+1 components of the vector P_i' here are as in the original vector P_i, and the last two components x_{i+1} and x_{i+2} are derived simultaneously by using two linear filters. The derivation process for x_{i+1} and x_{i+2} here will be described in details later. An example of expanding 2 vectors of $\{X_{i-1}\}$ into a total of 18 vectors $\{P_i'\}$ for the operations just described is shown in Fig. 5.22. Thus, having derived the corresponding m' vectors $\{P_i'\}$ from each of the m vectors $\{X_{i-1}\}$, the detector of system 8 then proceeds to evaluate the quantity C_i', by using eqn. 5.56, for each of the (m)(m') expanded vectors $\{P_i'\}$ here. The detected data symbol s_{i-n}' is then taken to have the value of x_{i-n} in the vector P_i' associated with the smallest C_i'. All vectors $\{P_i'\}$ for which $x_{i-n} \neq s_{i-n}$ are now discarded. The first component x_{i-n} and the last two components x_{i+1} and x_{i+2} of each of the remaining vectors $\{P_i'\}$ are omitted to give the corresponding n-component vectors $\{X_i\}$. Each vector X_i here is associated with the same C_i and C_i' as those of the corresponding vector P_i'. The detector

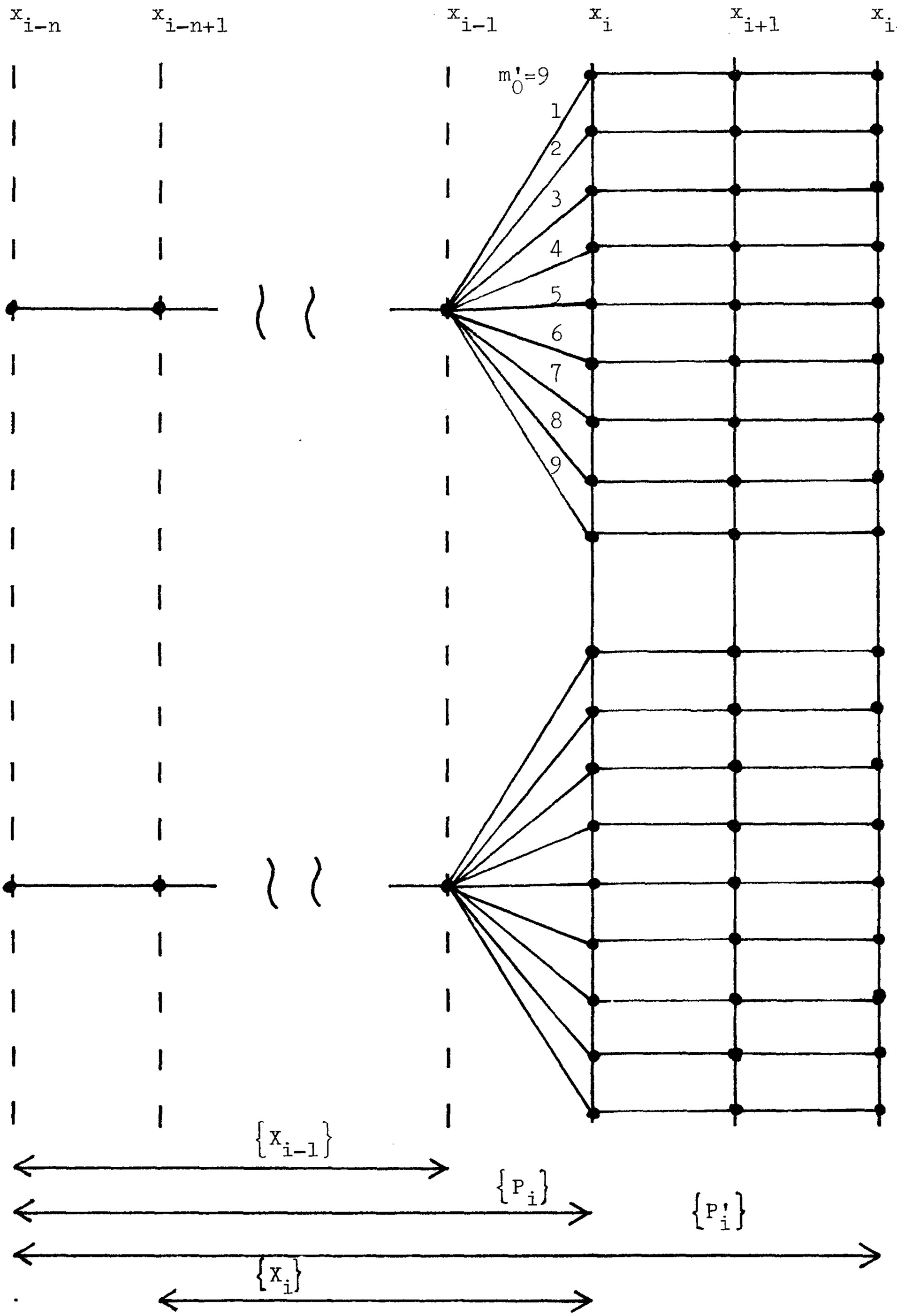


Fig. 5.22 Schematic diagram showing the expansion of vectors from $m\{x_{i-1}\}$ to $(m)(m')$ or $(m)(m_0')\{P_i'\}$ in the detection process of system 8. m' is as defined by eqn. 5.62. In this example, $m = 2$.

then proceeds with the selection of the m vectors $\{X_i\}$ associated with the smallest $\{C_i'\}$ subject to the constraint that only the m' vectors of $\{X_i\}$ originating from the best vector of X_{i-1} (having the smallest C_{i-1}' in the previous detection process) are available for the selection of the second vector of X_i here. These m selected vectors $\{X_i\}$ together with their associated costs $\{C_i\}$ are then stored and the detector is now ready for the next detection process where the detected value of the data symbol s_{i-n+1} will be determined in exactly the same way as is described above.

In the detection process described above, each n-component stored vector X_{i-1} is expanded into m' (n+3)-component vectors $\{P_i'\}$ (eqn. 5.80) having m' different values for the last 3 components x_i, x_{i+1}, and x_{i+2}. The derivation of the m' set of values for the 3 components x_i, x_{i+1}, and x_{i+2} here is carried out in system 8 by the use of two linear filters, and this derivation process is now described as below.

Three slightly different arrangements for the derivation process mentioned above have been developed and studied, and are described in the following three versions of system 8.

System 8 (Version a)

In the derivation of the m' set of values for the 3 components x_i, x_{i+1}, and x_{i+2} in the vectors $\{P_i'\}$ (eqn. 5.80), the detector here first removes from its input samples $r'_{i,0}, r'_{i,1}, \cdots, r'_{i,\alpha}$ (eqn. 5.52) detected values of all components involving the data symbols s_{i-n}, $s_{i-n+1}, \cdots, s_{i-1}$ whose values have already been determined (temporarily) as the values of the corresponding components $x_{i-n}, x_{i-n+1}, \cdots, x_{i-1}$. That is, the detector operates on its input sample values to give the

following complex-valued quantities

$$e'_0 = r'_{i,0} - \sum_{k=1}^{n} x_{i-k} y_k$$

$$e'_1 = r'_{i,1} - \sum_{k=1}^{n} x_{i-k} y_{k+1}$$

$$\vdots$$

$$e'_\alpha = r'_{i,\alpha} - \sum_{k=1}^{n} x_{i-k} y_{k+\alpha} \qquad (5.81)$$

where $\{y_j\}$ are the components of the channel sampled impulse response, and α is a constant to be described shortly. The detector then operates on the first p+1 sample values $e'_0, e'_1, \cdots, e'_p$ (eqn. 5.81), using the (p+1)-tap and the p-tap linear transversal filters shown in Fig. 5.23, to give the following two complex-valued quantities

$$f'_1 = e'_0 y^*_0 + e'_1 y^*_1 + \cdots + e'_p y^*_p \qquad (5.82)$$

and

$$f'_2 = e'_1 y^*_0 + \cdots + e'_p y^*_{p-1} \qquad (5.83)$$

where y^*_j is the complex conjugate of y_j. The value of p here is related to the value of α by

$$\alpha = p + 1 \qquad (5.84)$$

Having evaluated the two quantities f'_1 and f'_2, the detector then operates on the last p+1 sample values $e'_1, e'_2, \cdots, e'_{p+1}$ in eqn. 5.81, using the same two transversal filters shown in Fig. 5.23, to give

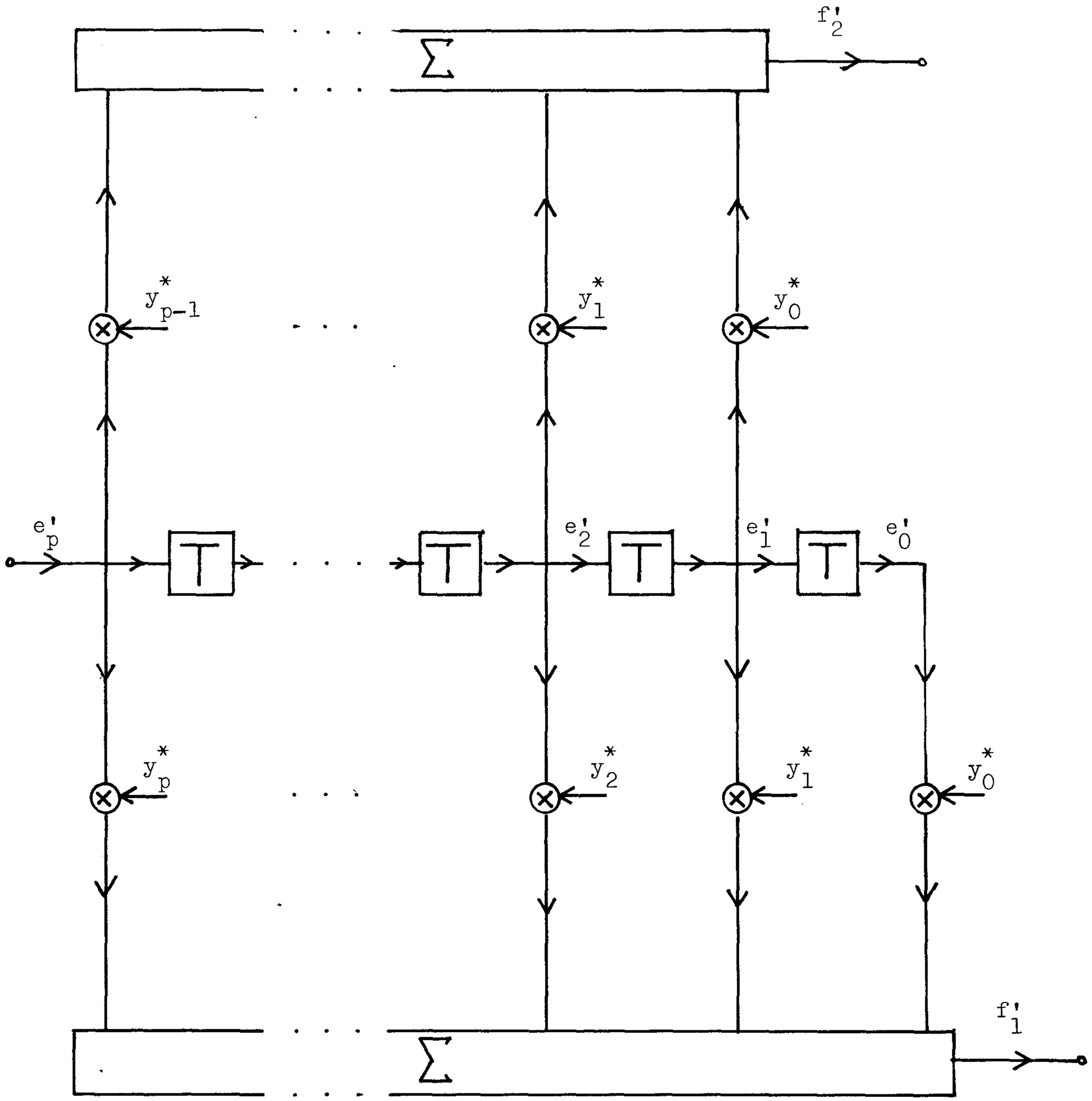


Fig. 5.23 Linear transversal filters used in system 8 to evaluate the quantities f'_1 and f'_2 (eqns. 5.82 and 5.83) in deriving the possible values for the component x_i in the expanded vectors $\{P'_i\}$ (eqn. 5.80).

the following two complex-valued quantities

$$f'_3 = e'_1 y^*_0 + e'_2 y^*_1 + \cdots + e'_{p+1} y^*_p \qquad (5.85)$$

and

$$f'_4 = e'_2 y^*_0 + \cdots + e'_{p+1} y^*_{p-1} \qquad (5.86)$$

The detector now operates on f'_1 and f'_2 to derive the m' values for the component x_i in P'_i (eqn. 5.80), and it then operates on f'_3 and f'_4 to derive the corresponding m' pairs of values for the components x_{i+1} and x_{i+2} in P'_i here. The detector here assumes that the values of $r'_{i,0}, r'_{i,1}, \cdots, r'_{i,\alpha}$ in eqn. 5.81 are as given by eqn. 5.53, and that $x_j = s_j$ for $j = i-n, i-n+1, \cdots, i-1$ so that eqn. 5.81 now becomes

$$\begin{aligned}
e'_0 &= s_i y_0 + w_i \\
e'_1 &= s_i y_1 + s_{i+1} y_0 + w_{i+1} \\
&\vdots \\
e'_\alpha &= s_i y_\alpha + s_{i+1} y_{\alpha-1} + \cdots + s_{i+} y_0 + w_{i+\alpha}
\end{aligned} \qquad (5.87)$$

where α is as given by eqn. 5.84, and $\{w_j\}$ are the noise components whose real and imaginary parts are statistically independent Gaussian random variables with zero mean and variance σ^2. Substituting eqn. 5.87 into eqns. 5.82, 5.83, 5.85, and 5.86, and after rearranging of terms, we arrive at

$$f'_1 = s_i a'_0 + s_{i+1} a'_1 + s_{i+2} a'_2 + \cdots + s_{i+p} a'_p + w'_1 \quad (5.88)$$

$$f'_2 = s_i b'_{-1} + s_{i+1} b'_0 + s_{i+2} b'_1 + \cdots + s_{i+p} b'_{p-1} + w'_2 \quad (5.89)$$

$$f'_3 = s_i a'_{-1} + s_{i+1} a'_0 + s_{i+2} a'_1 + \cdots + s_{i+p+1} a'_p + w'_3 \quad (5.90)$$

$$f'_4 = s_i b'_{-2} + s_{i+1} b'_{-1} + s_{i+2} b'_0 + \cdots + s_{i+p+1} b'_{p-1} + w'_4 \quad (5.91)$$

where

$$a'_k = \sum_{j=0}^{p-k} y_j y^*_{j+k} \quad (5.92)$$

$$a'_{-k} = \sum_{j=0}^{p} y_{j+k} y^*_j \quad (5.93)$$

$$b'_k = \sum_{j=0}^{p-k-1} y_j y^*_{j+k} \quad (5.94)$$

$$b'_{-k} = \sum_{j=0}^{p-1} y_{j+k} y^*_j \quad (5.95)$$

for any non-negative integer value of k, and the noise components are

$$w'_1 = \sum_{j=0}^{p} w_{i+j} y^*_j \quad (5.96)$$

$$w'_2 = \sum_{j=0}^{p-1} w_{i+1+j} y^*_j \quad (5.97)$$

$$w'_3 = \sum_{j=0}^{p} w_{i+1+j} y^*_j \quad (5.98)$$

$$w'_4 = \sum_{j=0}^{p-1} w_{i+2+j} y^*_j \quad (5.99)$$

Having stated the assumptions of eqns. 5.87 - 5.99 for the detection process here, the derivation process for x_i (using f'_1 and f'_2), x_{i+1} and x_{i+2} (using f'_3 and f'_4) is now described as follows. The detector

first operates on f'_1 and f'_2 (eqns. 5.82 and 5.83) to give the complex-valued quantity

$$x'_i = v\,(f'_1 b'_0 - f'_2 a'_1) \qquad (5.100)$$

where $$v = (a'_0 b'_0 - a'_1 b'_{-1})^{-1} \qquad (5.101)$$

and a'_0, a'_1, b'_0, b'_{-1} here are as given by eqns. 5.92 - 5.95. It should be noted that, v is a real-valued quantity since a'_0 and b'_0 are real-valued quantities and a'_1 and b'_{-1} are a complex conjugate pair, as can be seen from eqns. 5.92 - 5.95. It can be seen from eqns. 5.88 and 5.89 that, the quantity x'_i (eqn. 5.100) may be reduced to

$$x'_i = s_i + J_i + u'_i \qquad (5.102)$$

where the intersymbol interference component J_i and the noise component u'_i here are given as follows

$$J_i = s_{i+2} v(a'_2 b'_0 - b'_1 a'_1) + \cdots + s_{i+p} v(a'_p b'_0 - b'_{p-1} a'_1) \qquad (5.103)$$

$$u'_i = v\,(w'_1 b'_0 - w'_2 a'_1) \qquad (5.104)$$

The quantity x'_i here is treated as the estimate of the data symbol s_i whose value is to be determined as the value of x_i in P'_i (eqn. 5.80). It can be seen from eqns. 5.102 and 5.103 that, the intersymbol interference of s_{i+1} is removed in the estimated value x'_i of s_i. Thus, having evaluated the value of x'_i by using eqn. 5.100, the detector then operates on x'_i to give m' possible values of s_i for the component x_i in P'_i. This is carried out as follows. When m' = 4, Table 5.8 is used to derive the 4 possible values for x_i, and when m' = 9, Table 5.9 is used to

derive the 9 possible values for x_i, bearing in mind that x_i here may have any of the 16 possible complex values of the data symbol s_i. When $m' = 16$, however, all the 16 possible values of s_i are taken for the 16 values of x_i, and so there is actually no need to evaluate any of the quantities f'_1, f'_2, and x'_i in this case. Having derived the m' possible values for the component x_i, the detector then proceeds to operate on f'_3 and f'_4 (eqns. 5.85 and 5.86) to give, for each value of x_i just derived, the complex-valued quantity

$$x'_{i+1} = v (f'_3 b'_0 - f'_4 a'_1) - x_i v' \qquad (5.105)$$

where $$v' = v (a'_{-1} b'_0 - b'_{-2} a'_1) \qquad (5.106)$$

and a'_1, a'_{-1}, b'_0, b'_{-2} are as given by eqns. 5.92 - 5.95, and v is as given by eqn. 5.101. Assuming that $x_i = s_i$ here, it can be seen from eqns. 5.90 and 5.91 that the quantity x'_{i+1} in eqn. 5.105 may be reduced to

$$x'_{i+1} = s_{i+1} + J_{i+1} + u'_{i+1} \qquad (5.107)$$

where the intersymbol interference component J_{i+1} and the noise component u'_{i+1} here are given as

$$J_{i+1} = s_{i+3} v(a'_2 b'_0 - b'_1 a'_1) + \cdots + s_{i+p+1} v(a'_p b'_0 - b'_{p-1} a'_1) \qquad (5.108)$$

$$u'_{i+1} = v (w'_3 b'_0 - w'_4 a'_1) \qquad (5.109)$$

which obviously resemble eqns. 5.102 - 5.104. The quantity x'_{i+1} is here treated as the estimate of the data symbol s_{i+1}. The detector now operates on x'_{i+1} (eqn. 5.105) according to Table 5.6 to give the

value for x_{i+1}, where x_{i+1} here may have any of the 16 possible values of the data symbol s_{i+1}. Having derived the value for x_{i+1} here, the detector operates on f'_4 (eqn. 5.91) to give the complex-valued quantity

$$x'_{i+2} = \frac{1}{b'_0} (f'_4 - x_i b'_{-2} - x_{i+1} b'_{-1}) \qquad (5.110)$$

where b'_0, b'_{-1}, and b'_{-2} are as given by eqn. 5.95. Assuming that $x_i = s_i$ and $x_{i+1} = s_{i+1}$, it can be seen from eqn. 5.91 that the quantity x'_{i+2} evaluated here (eqn. 5.110) may be reduced to

$$x'_{i+2} = s_{i+2} + J_{i+2} + u'_{i+2} \qquad (5.111)$$

where the intersymbol interference component J_{i+2} and the noise component u'_{i+2} here are given by

$$J_{i+2} = \frac{1}{b'_0} (s_{i+3} b'_1 + s_{i+4} b'_2 + \cdots + s_{i+p+} b'_{p-1}) \qquad (5.112)$$

$$u'_{i+2} = \frac{w'_4}{b'_0} \qquad (5.113)$$

Again, the quantity x'_{i+2} here is treated as the estimate of the data symbol s_{i+2}. The detector now operates on x'_{i+2} (eqn. 5.110) according to Table 5.6 to give the value for x_{i+2}, where x_{i+2} here has a possible value of s_{i+2}. Thus, for each of the m' values derived for x_i, the detector derives one value for x_{i+1} (using eqn. 5.105 and Table 5.6) and one value for x_{i+2} (using eqn. 5.110 and Table 5.6), to give the corresponding m' (n+3)-component vectors $\{P'_i\}$ (eqn. 5.80 and Fig. 5.22). All these are repeated for each of the m stored vectors $\{X_{i-1}\}$ (eqn. 5.8 and Fig. 5.22), to give a total of (m)(m') vectors $\{P'_i\}$ here. The detector then proceeds with the evaluations of $\{C'_i\}$ (eqn. 5.56) for these $\{P'_i\}$ and so on as is described before.

System 8 (Version b)

In version a of system 8, the components x_{i+1} and x_{i+2} in the vectors $\{P'_i\}$ (eqn. 5.80) are restricted to have the possible values of the corresponding 16-point QAM data symbols. This restriction is removed in version b of system 8. The detection process involved in version b here is otherwise the same as that involved in version a of system 8. Thus, in the derivation of the m' sets of values for the 3 components x_i, x_{i+1}, and x_{i+2} in the expanded vectors $\{P'_i\}$ (eqn. 5.80 and Fig. 5.22), the detector in system 8 with version b operates on its input sample values (eqn. 5.52) to give the complex-valued quantities given by eqn. 5.81. This detector then operates on these sample values (eqn. 5.81), using the linear filters shown in Fig. 5.23, to give the four complex-valued quantities f'_1, f'_2, f'_3, and f'_4 (eqns. 5.82, 5.83, 5.85, and 5.86). The estimated value x'_i of the data symbol s_i is then evaluated from f'_1 and f'_2 by using eqn. 5.100. Table 5.8 or 5.9 with this value of x'_i is then used to derive m' values for the component x_i, where m' here may have a value of 4 or 9. When m' = 16, all the 16 possible values of s_i are taken for the 16 values of x_i and in this case, it is not necessary to evaluate the values of f'_1, f'_2, and x'_i. The operation described so far for version b is exactly the same as that for version a . Having derived the m' values for x_i, the detector of version b next evaluates, for each value of x_i just derived, an estimated value x'_{i+1} of s_{i+1} by using eqn. 5.105. The value of x_{i+1} is then taken to be the value of x'_{i+1} here with the constraint that

$$\left|\mathrm{Re}(x_{i+1})\right| \leq 3$$

$$\left|\mathrm{Im}(x_{i+1})\right| \leq 3 \qquad (5.114)$$

where Re(.) and Im(.) are the real and imaginary parts of (.) respectively. That is, $\mathrm{Re}(x_{i+1}) = \mathrm{Re}(x'_{i+1})$ unless $\mathrm{Re}(x'_{i+1}) > 3$ (or < -3) when $\mathrm{Re}(x_{i+1}) = 3$ (or -3), and $\mathrm{Im}(x_{i+1}) = \mathrm{Im}(x'_{i+1})$ unless $\mathrm{Im}(x'_{i+1}) > 3$ (or < -3) when $\mathrm{Im}(x_{i+1}) = 3$ (or -3), where 3 and -3 are the maximum and minimum possible values respectively for the real or imaginary part of the data symbol s_{i+1}. Having determined the value of x_{i+1}, the detector then evaluates the estimated value x'_{i+2} of s_{i+2} by using eqn. 5.110. The value of x_{i+2} is then taken to be the value of x'_{i+2} here with the constraint that

$$\begin{aligned} |\mathrm{Re}(x_{i+2})| &\leq 3 \\ |\mathrm{Im}(x_{i+2})| &\leq 3 \end{aligned} \qquad (5.115)$$

which resembles the constraint of (5.114) for the component x_{i+1}. Having derived the m' sets of values for the components x_i, x_{i+1}, and x_{i+2} in the corresponding m' vectors $\{P'_i\}$ (eqn. 5.80) originating from one of the m stored vectors $\{X_{i-1}\}$ (eqn. 5.8), the detector then repeats the whole process here for each of the remaining m-1 vectors $\{X_{i-1}\}$ to give a total of (m)(m') expanded vectors $\{P'_i\}$ having the (m)(m') sets of values for the corresponding components x_i, x_{i+1}, and x_{i+2}. The detector then proceeds with the evaluations of $\{C'_i\}$ (eqn. 5.56) for these $\{P'_i\}$ and so on as is described before.

Thus, in deriving the value for any of the components x_{i+1} and x_{i+2} in the vector P'_i, the detector here uses no prior knowledge of the possible values (3, 1, -1, and -3) of the real and imaginary parts of the data symbols s_{i+1} and s_{i+2}, except that the maximum and minimum possible values here are given by 3 and -3 respectively. The reason for not setting the value of $\mathrm{Re}(x_h)$ or $\mathrm{Im}(x_h)$ (h = i+1, i+2) to a possible

value of $Re(s_h)$ or $Im(s_h)$ when $Re(x'_h)$ or $Im(x'_h)$ has a value that lies between 3 to -3 in the detection process here (version b) is given as follows. Consider first the case when the estimated value $Re(x'_h)$ of $Re(s_h)$ ($h = i+1, i+2$) lies between 3 to -3. The distances from this value of $Re(x'_h)$ to the nearest two possible values of $Re(s_h)$ now differ by an amount of less than 2. This difference can in fact be very small because of the presence of the intersymbol interference component $Re(J_h)$ (eqn. 5.107 or 5.111) in $Re(x'_h)$. Thus, the distances from $Re(x'_h)$ to the nearest two possible values of $Re(s_h)$ can differ by an amount so small that it is now very likely for the incorrect possible value of $Re(s_h)$ to be taken as the value of $Re(x_h)$ if $Re(x_h)$ is restricted to have a possible value of $Re(s_h)$. Furthermore, if $Re(x_h)$ has the incorrect possible value of $Re(s_h)$, then the distance between the values of $Re(x_h)$ and $Re(s_h)$ is equal to 2 here and is larger than the distance between the values of $Re(x'_h)$ and $Re(s_h)$, bearing in mind that it is required here to determine the value of $Re(s_h)$ as the value of $Re(x_h)$. Consequently, it appears that because of the presence of the intersymbol interference component $Re(J_h)$ in $Re(x'_h)$, a better tolerance to noise may be achieved by taking the value of $Re(x_h)$ as the value of $Re(x'_h)$ when $Re(x'_h)$ lies between 3 to -3. The same argument just described can, of course, be applied to the case when the estimated value $Im(x'_h)$ of $Im(s_h)$ ($h = i+1, i+2$) lies between 3 to -3, and hence a better tolerance to noise may be achieved by taking x_h to have the value of x'_h instead of a possible value of s_h here. If, however, $Re(x'_h)$ or $Im(x'_h)$ has a value larger than 3 (or smaller than -3), then the value of $Re(s_h)$ or $Im(s_h)$ is always nearer to the value of $Re(x_h)$ or $Im(x_h)$ than to the value of $Re(x'_h)$ or $Im(x'_h)$ so long as $Re(x_h)$ or $Im(x_h)$ is taken to have the value of 3 (or -3). In this case, a better tolerance to noise is likely to be achieved by having the constraints of (5.114) and (5.115).

System 8 (Version c)

System 8 with version c is a modification of system 8 with version b. The modification here is aimed at reducing the number of operations involved in deriving the values for the components x_i, x_{i+1}, and x_{i+2} in the vectors $\{P'_i\}$ (eqn. 5.80 and Fig. 5.22) in the detection process of system 8 with version b. Basically this is achieved here by modifying the linear transversal filters (Fig. 5.23) used to derive the values for x_i, x_{i+1}, and x_{i+2} in version b of system 8 to have fewer taps. This version is otherwise the same as version b of system 8.

Thus, in the derivation of the m' sets of values for the components x_i, x_{i+1}, and x_{i+2} in the expanded vectors $\{P'_i\}$ (Fig. 5.22), the detector in system 8 with version c operates on its input sample values $\{r'_{i,h}\}$ (eqn. 5.52) to give the complex-valued quantities $\{e'_h\}$ given by eqn. 5.81. The detector then operates on $\{e'_h\}$, using the linear filters shown in Fig. 5.24, to give the following four complex-valued quantities

$$f'_1 = e'_f y^*_f + e'_{f+1} y^*_{f+1} + \cdots + e'_p y^*_p \qquad (5.116)$$

$$f'_2 = e'_{f+1} y^*_f + \cdots + e'_p y^*_{p-1} \qquad (5.117)$$

$$f'_3 = e'_{f+1} y^*_f + e'_{f+2} y^*_{f+1} + \cdots + e'_{p+1} y^*_p \qquad (5.118)$$

$$f'_4 = e'_{f+2} y^*_f + \cdots + e'_{p+1} y^*_{p-1} \qquad (5.119)$$

where y^*_j is the complex conjugate of y_j, y_f is the first significant component of the channel sampled impulse response, and p and p+1 are the number of taps used in the two linear filters of Fig. 5.23. It can be seen from Figs. 5.23 and 5.24 that, the two linear filters used in this version of system 8 can be obtained from the corresponding two linear filters used in version b of system 8 by removing those

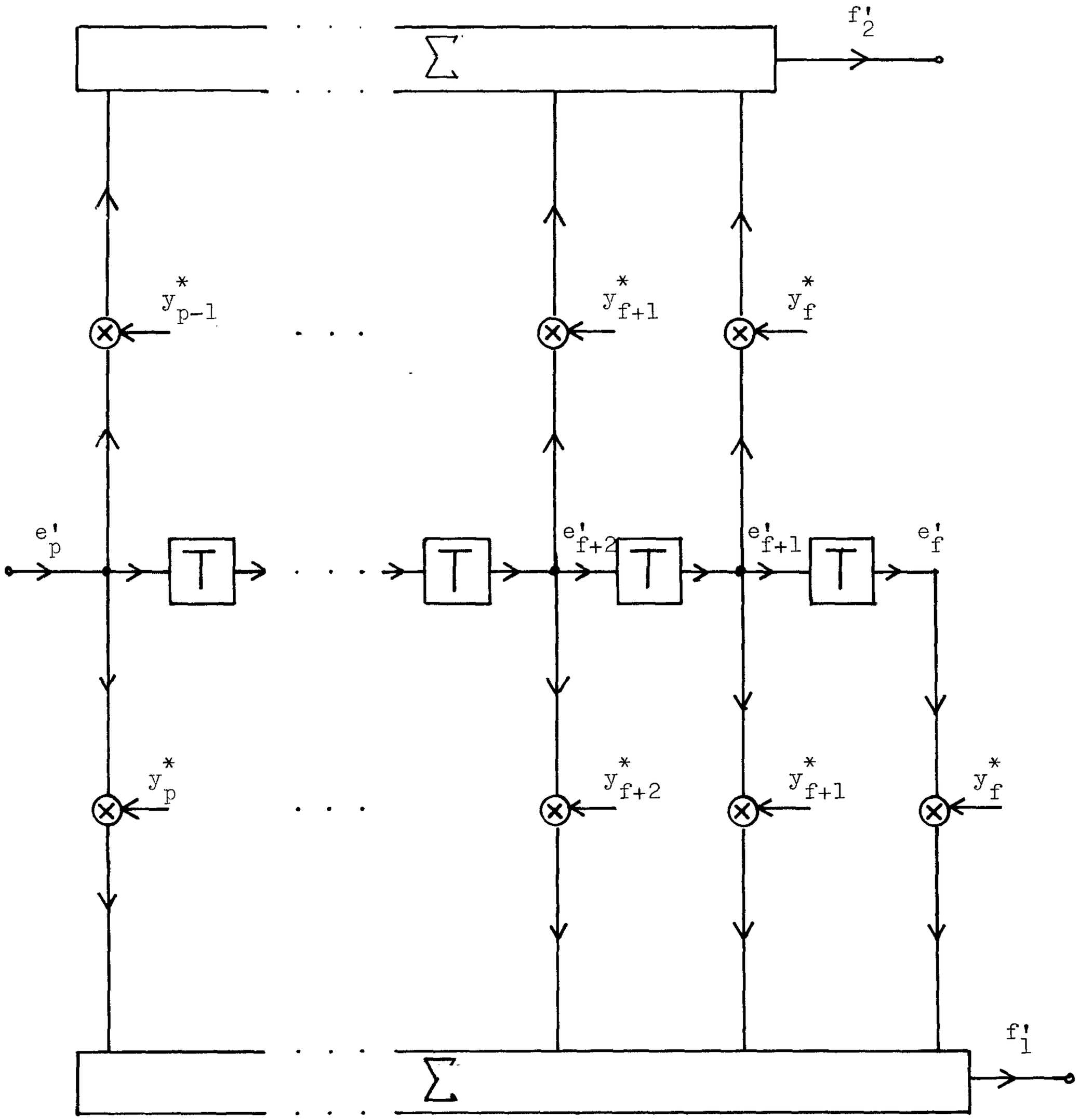


Fig. 5.24 Linear transversal filters used in system 8 (version c) to evaluate the quantities f'_1 and f'_2 (eqns. 5.116 and 5.117) in deriving the possible values for the component x_i in the expanded vectors $\{P'_i\}$ (eqn. 5.80).

taps associated with the components $y_0, y_1, \cdots, y_{f-1}$, bearing in mind that these components are assumed in the detection process here to have negligible magnitudes. Thus, the number of complex-valued multiplications involved in the linear filters to evaluate the quantities f'_1, f'_2, f'_3, and f'_4 for each of the m stored vectors $\{X_{i-1}\}$ (eqn. 5.8 and Fig. 5.22) is reduced from $4p+2$ in version b of system 8 to $4(p-f)+2$ in version c of system 8, bearing in mind that $f > 0$ here. By substituting the values of $\{e'_h\}$ in eqn. 5.87 into eqns. 5.116 - 5.119, and after rearranging of terms, the quantities f'_1, f'_2, f'_3, and f'_4 can be reduced to those given by eqns. 5.88 - 5.91 where the various constants and noise components given by eqns. 5.92 - 5.99 are here given by

$$a'_k = \sum_{j=f-k}^{p-k} y_j y^*_{j+k} \tag{5.120}$$

$$a'_{-k} = \sum_{j=f}^{p} y_{j+k} y^*_j \tag{5.121}$$

$$b'_k = \sum_{j=f-k}^{p-k-1} y_j y^*_{j+k} \tag{5.122}$$

$$b'_{-k} = \sum_{j=f}^{p-1} y_{j+k} y^*_j \tag{5.123}$$

where $y_h = 0$ for $h < 0$, and

$$w'_1 = \sum_{j=f}^{p} w_{i+j} y^*_j \tag{5.124}$$

$$w'_2 = \sum_{j=f}^{p-1} w_{i+1+j} y^*_j \tag{5.125}$$

$$w'_3 = \sum_{j=f}^{p} w_{i+1+j} y^*_j \tag{5.126}$$

$$w'_4 = \sum_{j=f}^{p-1} w_{i+2+j} y^*_j \tag{5.127}$$

Having evaluated the values of f'_1, f'_2, f'_3, and f'_4 (by using eqns. 5.116 - 5.119), the detection process of system 8 with version c then proceeds with the derivation of the m' sets of values for the components x_i, x_{i+1}, and x_{i+2} in $\{P'_i\}$ (Fig. 5.22) and the evaluation of $\{C'_i\}$ (eqn. 5.56) for these $\{P'_i\}$, exactly as for system 8 with version b, bearing in mind that the values of f'_1, f'_2, f'_3, f'_4, $\{a'_k\}$, $\{a'_{-k}\}$, $\{b'_k\}$, and $\{b'_{-k}\}$ are here given by eqns. 5.116 - 5.123.

Thus, system 8 with version c differs from system 8 with version b only in having fewer taps for the linear transversal filters (Figs. 5.23 and 5.24) used to derive the values for x_i, x_{i+1}, and x_{i+2} in $\{P'_i\}$ (Fig. 5.22), and in having fewer terms in the constants $\{a'_k\}$ $\{a'_{-k}\}$, $\{b'_k\}$, and $\{b'_{-k}\}$ used to evaluate the estimated values x'_i, x'_{i+1}, and x'_{i+2} (eqns. 5.100, 5.105, and 5.110). The number of operations involved in this version of system 8 is therefore fewer than that involved in version b of system 8.

Computer simulation tests have been carried out to determine the tolerances to Gaussian noise of system 8 with versions a, b, and c operating over channel A (Table 2.4). Various values of p and f have been tested here and the more promising results are shown in Fig. 5.25, where f is the number of small components at the front end of the channel sampled impulse response, and p and p+1 are the number of taps used in the linear transversal filters shown in Fig. 5.23. The 95% confidence limits of the results shown in Fig. 5.25 are about $\pm$0.5 dB. The bit error rate and the signal to noise ratios here are as defined by eqns. 5.3 and 5.4 respectively. Fig. 5.25 also shows the performances of the optimum nonlinear equalizer described in section 2.6.3 and the system described in section 5.3.

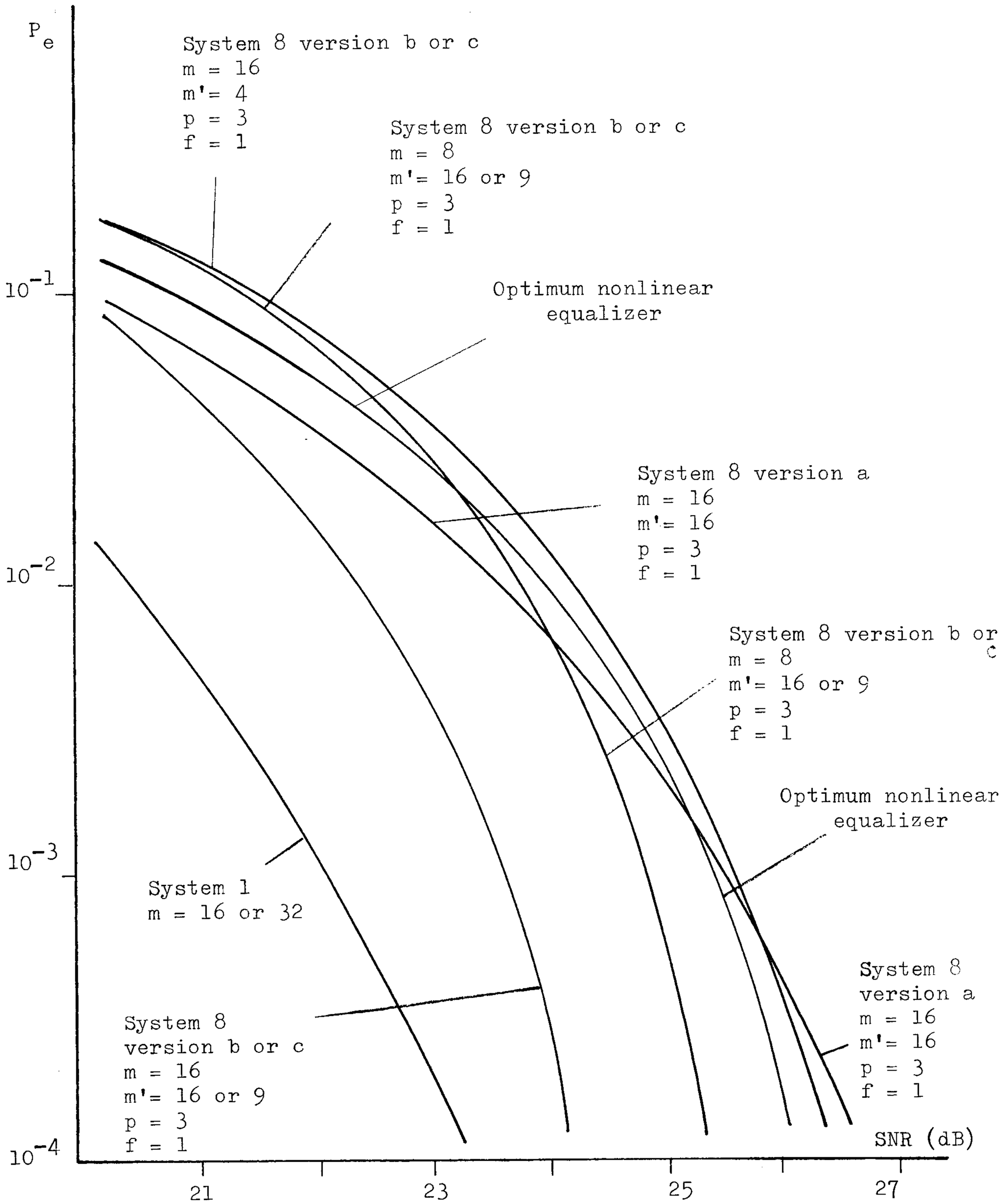


Fig. 5.25 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 8 operating over channel A (Table 2.4). Number of components in each stored vector X_i is (n) = 16. m' is the number of expanded vectors for each of the m stored vectors. p and f are as shown in Fig. 5.23 and 5.24.

It can be seen from Fig. 5.25 that, the tolerance to noise of system 8 with version a is slightly better than that of the optimum nonlinear equalizer at low signal to noise ratios where the error rates are higher than 10^{-3}. At error rates below 10^{-3}, this version of system 8 appears to have an inferior performance as compared to that of the equalizer. System 8 with version a therefore does not appear to be promising when operating over channel A here, bearing in mind that this system involves a more complicated detection process in relation to that involved in the equalizer.

As it appears in Fig. 5.25, system 8 with version c has the same tolerance to noise as that of the system with version b for given values of m, m', f, and p, where m is the number of stored vectors $\{X_{i-1}\}$ (eqn. 5.8), m' is the number of vectors $\{P'_i\}$ (eqn. 5.80) expanded from each of these stored vectors (Fig. 5.22), f is the number of small components at the front end of the channel sampled impulse response, and p and p+1 are the number of taps of the filters shown in Fig. 5.23. When m = 16, the performance of system 8 with version b or c remains unchanged as the value of m' is reduced from 16 to 9, and the improvement in tolerance to noise over the optimum nonlinear equalizer here is about 2 dB at an error rate of 10^{-4}, the improvement being larger as the error rate decreases further. When m' is reduced to 4, the degradation in the performance here becomes large and system 8 with version b or c now has the same tolerance to noise as that of the equalizer. When m = 8, there is again no difference in the performance of system 8 with version b or c for m' = 16 or 9. The loss in the tolerance to noise that results when m is reduced from 16 to 8 here, is about 1 dB for the range of signal to noise ratios considered in Fig. 5.25.

It can be seen from the results just described that, system 8 with version c is the most promising of the systems studied here, bearing in mind that this version of system 8 involves fewer operations in the detection process as compared to that involved in version b of system 8. Both versions b and c of system 8 appear to be able to achieve a better tolerance to noise over channel A here in relation to that of the optimum nonlinear equalizer without involving an excessive number of operations per data symbol in the detection process. For example, when system 8 with version b or c operates with $m = 16$ and $m' = 9$, the improvement gained in tolerance to noise over the equalizer here is about 2 dB at an error rate of 10^{-4}. The number of measurements of the quantities $\{C_i'\}$ (eqn. 5.56) per data symbol required in system 8 here is $(m)(m')$ or 144. The additional operations per data symbol involved in deriving the components x_i, x_{i+1}, and x_{i+2} in the expanded vectors $\{P_i'\}$ (eqn. 5.80 and Fig. 5.22) here can also be seen from the descriptions of the system given previously to be of the same order of magnitude as that involved in the measurements of $\{C_i'\}$. It should perhaps be reminded that system 8 has the advantage over system 1 and optimum nonlinear equalizer that, it can operate directly over the received sample values at the channel output (Fig. 2.3) and therefore without requiring the sampled impulse response of the channel to be adjusted into any particular form by means of a linear filter. System 8 with version b or c therefore appears to be very promising when operating over channel A here.

5.11 Further Computer Simulation Results

Previous sections have shown that system 8 with version b or c is able to achieve a satisfactory performance over channel A (Table 2.4). Further computer simulation tests have therefore been carried out on this system operating over channels B, C, D, E, and F (Tables 2.4 and 2.5) to justify its usefulness over a wider range of conditions. The characteristics of these channels are described in section 2.5. Thus, channel A introduces very severe group-delay distortion as well as severe attenuation distortion, channel B introduces very severe attenuation distortion, and channels C, D, E, and F all introduce typical levels of both attenuation and group-delay distortions. In the computer simulation tests here, 10,000 data symbols $\{s_i\}$ are transmitted for each measurement of the bit error rate, and the results are plotted as bit error rate (eqn. 5.3) versus signal to noise ratio (eqn. 5.4). When the results are more scattered (which normally occur at low error rates), more tests are carried out to measure the bit error rates at the same signal to noise ratios here. The results of these tests are shown in Figs. 5.26 - 5.30 where the 95% confidence limits of the curves are about ± 0.5 dB. The performances of the optimum nonlinear equalizer described in section 2.6.3 and the system 1 described in section 5.3 are also included here.

Various values of f and p used in the detection process of system 8 have been considered in the computer simulation tests here, but only their values giving the better results are considered in Figs. 5.26 - 5.30. The value of n used in the tests here is 16, where n is the number of components of a stored vector X_i (eqn. 5.8) in system 1 or 8.

All computer simulation tests here have been carried out on the CDC 7600 computer in Manchester, and the programs are written in FORTRAN. The computer program for system 8 with version b is shown in appendix B2.

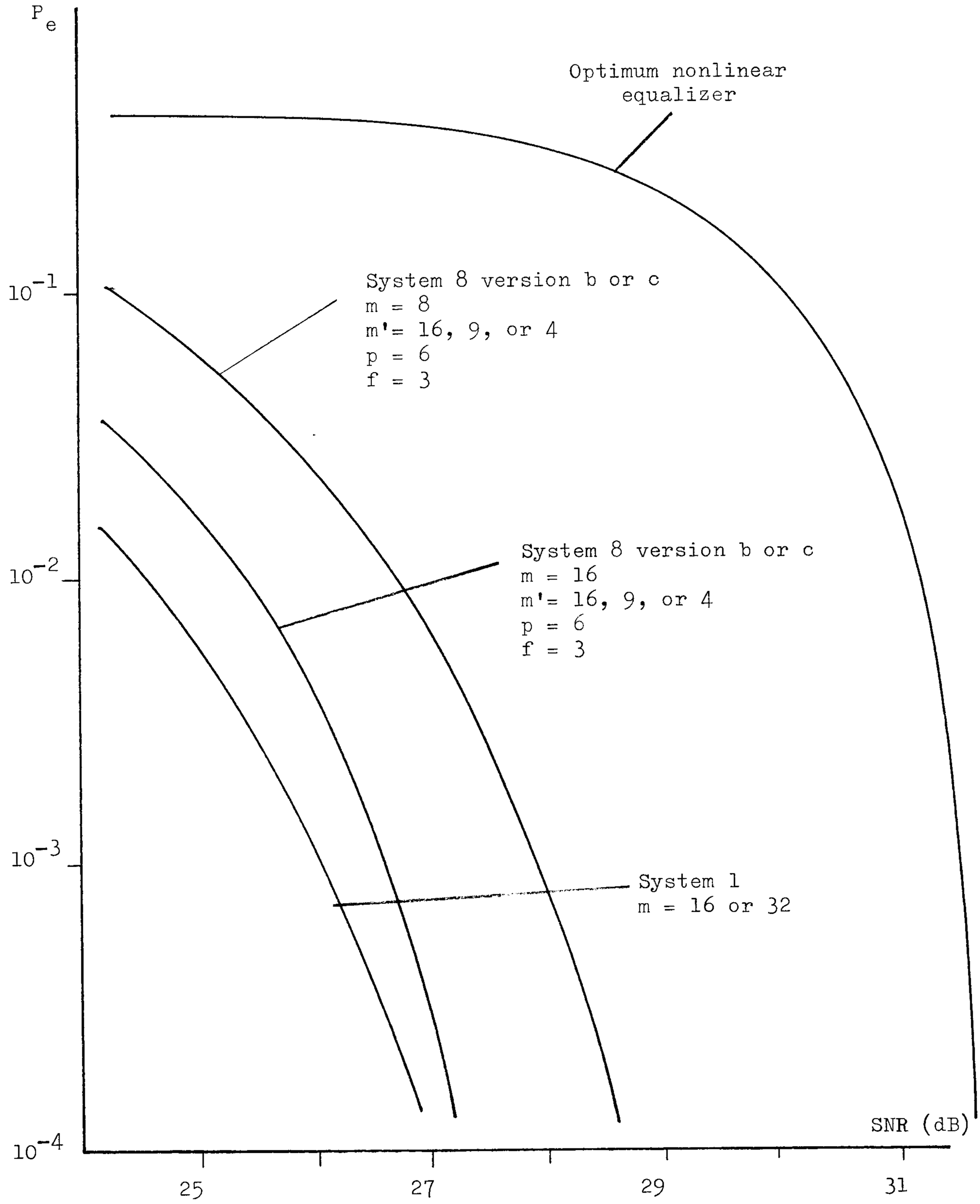


Fig. 5.26 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 8 operating over channel B (Table 2.4). Number of components in each stored vector X_i is (n) = 16. m' is the number of expanded vectors for each of the m stored vectors. p and f are as shown in Figs. 5.23 and 5.24.

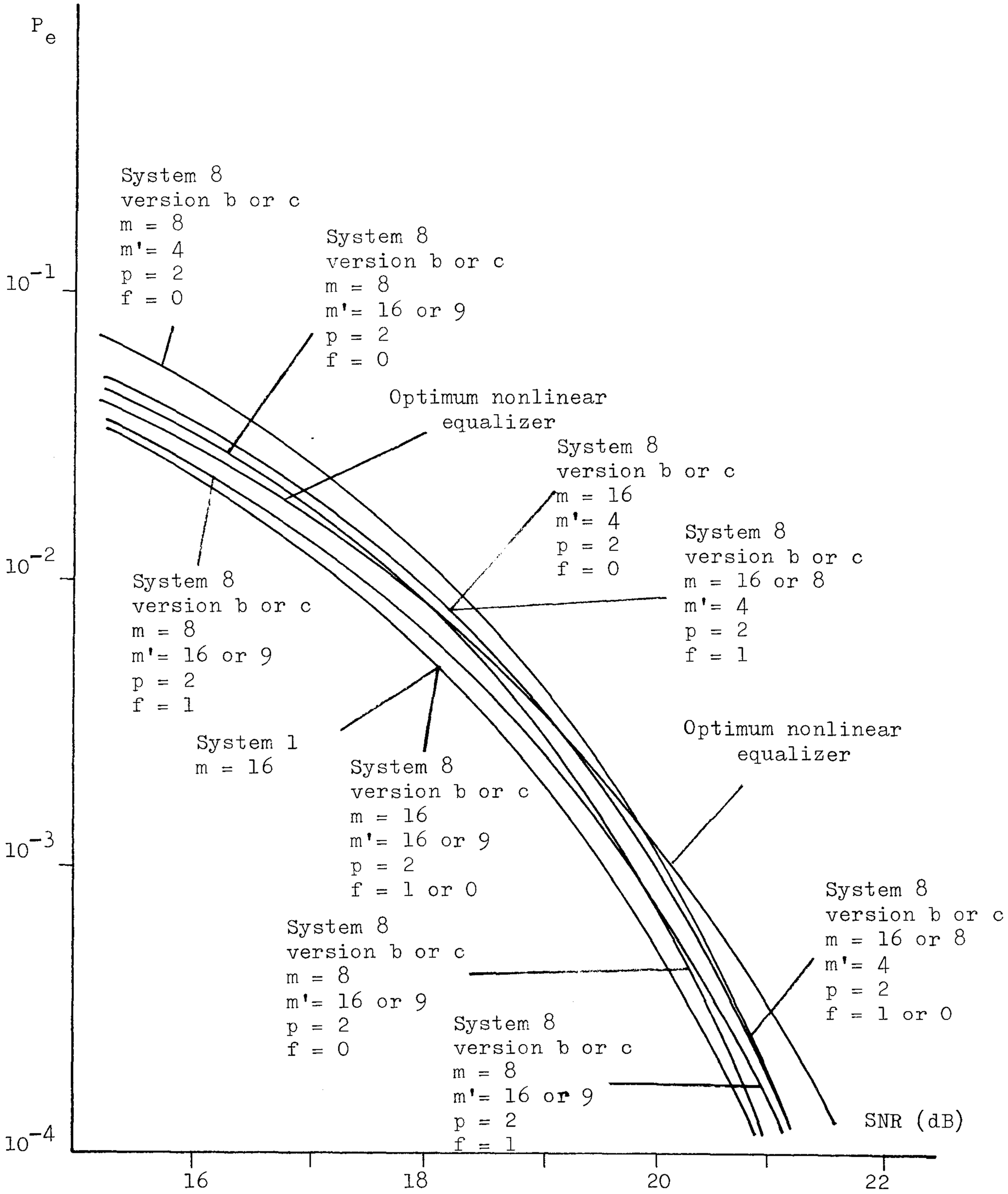


Fig. 5.27 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 8 operating over channel C (Table 2.4). Number of components in each stored vector X_i is (n) = 16. m' is the number of expanded vectors for each of the m stored vectors. p and f are as shown in Figs. 5.23 and 5.24.

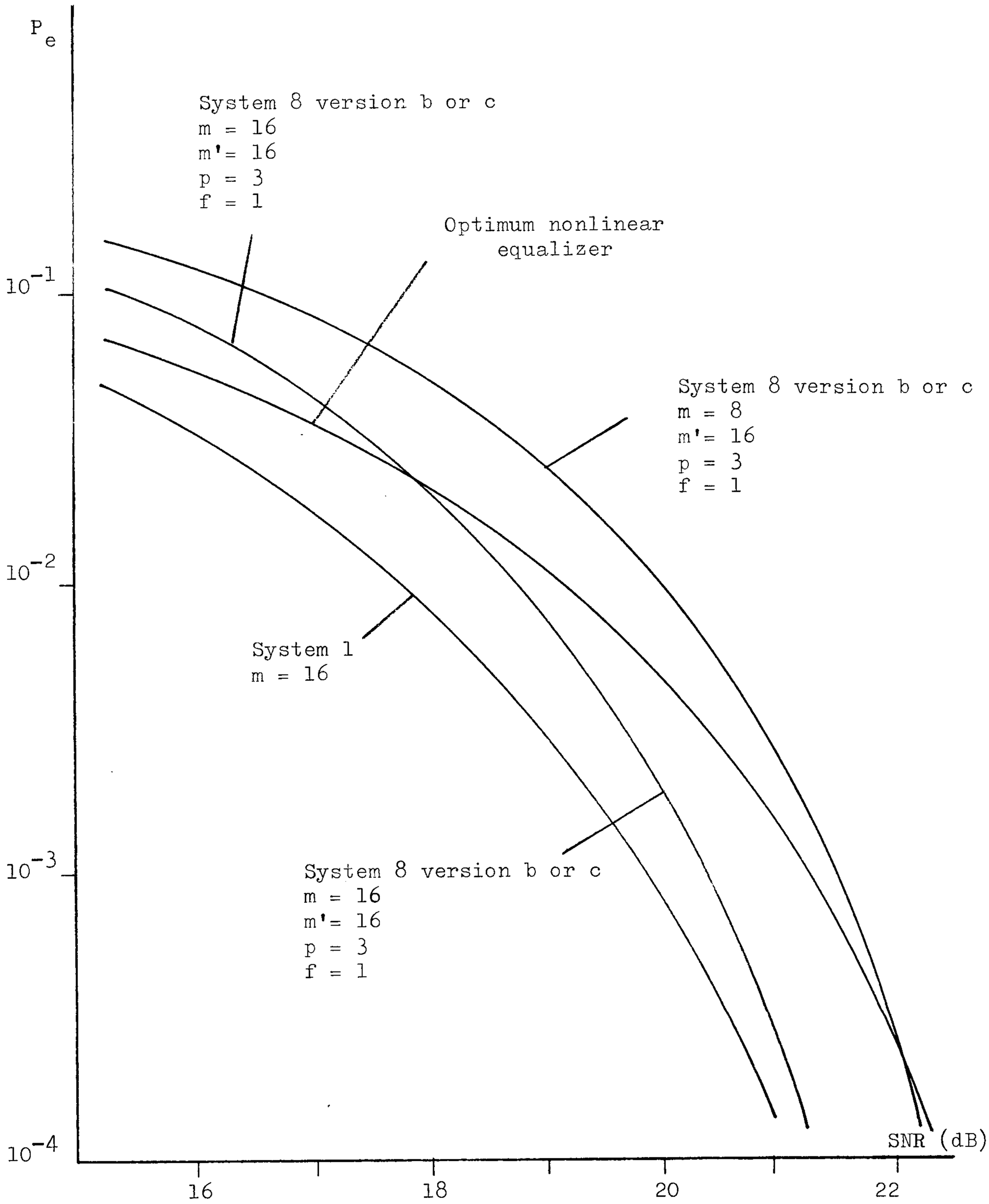


Fig. 5.28 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 8 operating over channel D (Table 2.5). Number of components in each stored vector X_i is (n) = 16. m' is the number of expanded vectors for each of the m stored vectors. p and f are as shown in Figs. 5.23 and 5.24.

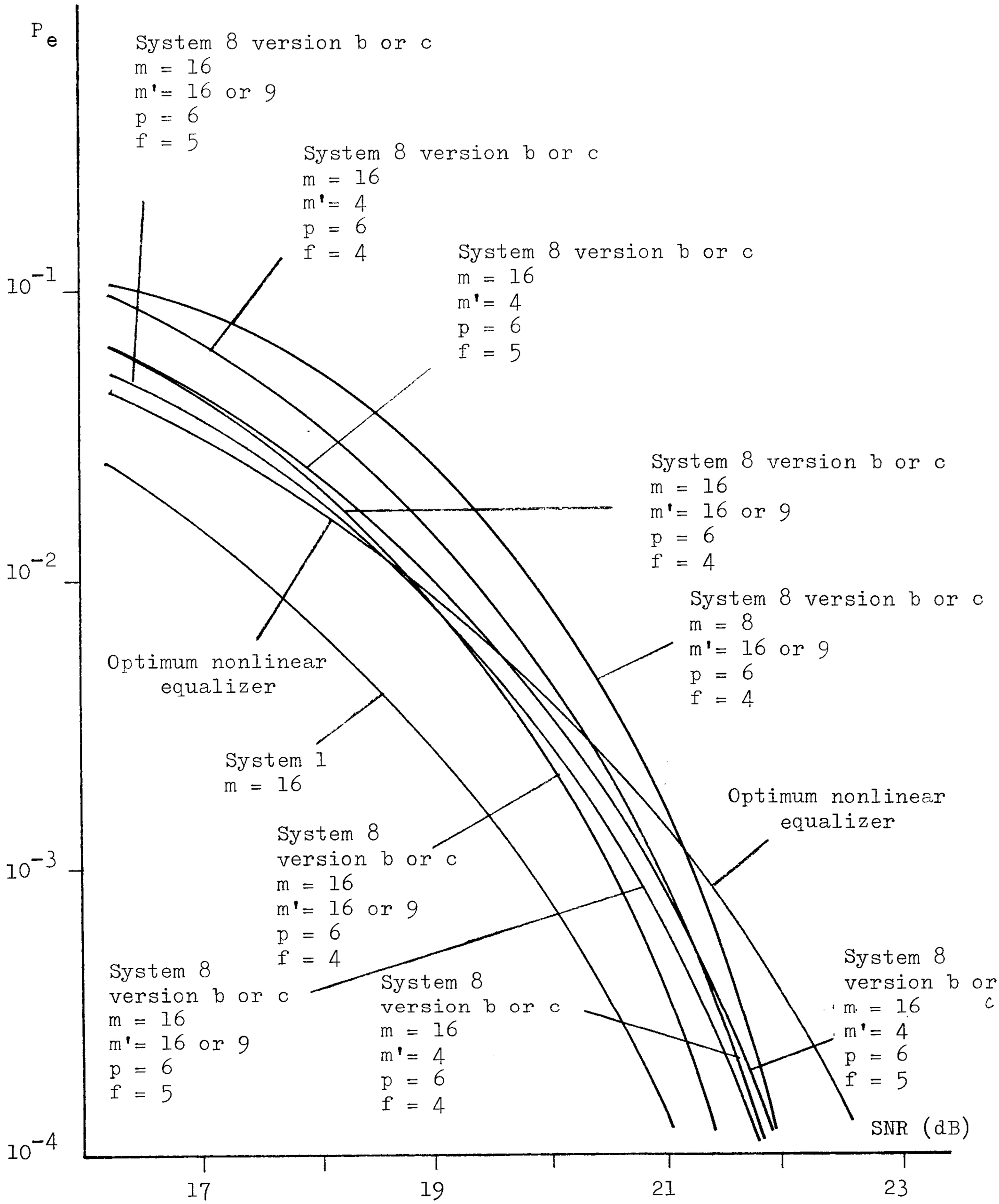


Fig. 5.29 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 8 operating over channel E (Table 2.5). Number of components in each stored vector X_i is (n) = 16. m' is the number of expanded vectors for each of the m stored vectors. p and f are as shown in Figs. 5.23 and 5.24.

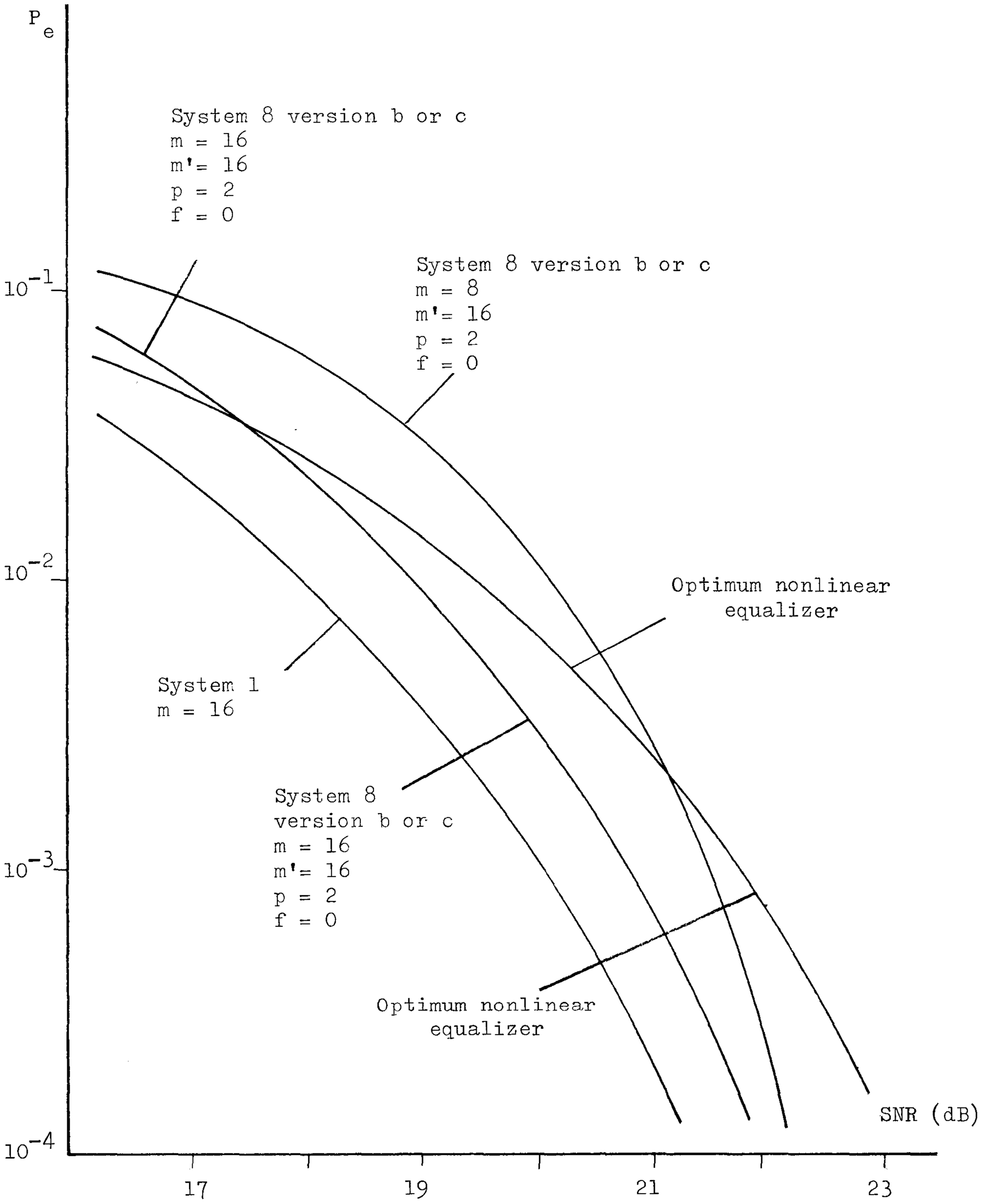


Fig. 5.30 Variation of error rate P_e (eqn. 5.3) with signal to noise ratio SNR (eqn. 5.4) for system 8 operating over channel F (Table 2.5). Number of components in each stored vector X_i is (n) = 16. m' is the number of expanded vectors for each of the m stored vectors. p and f are as shown in Figs. 5.23 and 5.24.

5.12 Assessment of Systems

The operations involved in the detection processes of system 1 and system 8 with versions b and c are summarised briefly as follows.

System 1 : A linear prefilter that acts as a 'whitened matched-filter' is inserted ahead of a near-maximum likelihood detector. The detector here expands each of the m n-component stored vectors $\{X_{i-1}\}$ (eqn. 5.8) into 16 (n+1)-component vectors $\{P_i\}$ (eqn. 5.9), and evaluates a cost value C_i (eqns. 5.10 - 5.13) for each of these $\{P_i\}$. The detected data symbol s'_{i-n} is then taken to have the value of the first component x_{i-n} in the vector P_i associated with the smallest cost. The m vectors $\{P_i\}$ associated with the smallest costs are then selected to give the m stored vectors $\{X_i\}$ having the last n components of the corresponding $\{P_i\}$ here.

System 8 (Version b) : The near-maximum likelihood detector here operates directly over the received sample values at the channel output (Fig. 2.3) without the use of any prefilter. The detector here expands each of the m n-component stored vectors $\{X_{i-1}\}$ (eqn. 5.8) into m' (n+3)-component vectors $\{P'_i\}$ (eqn. 5.80), where m' may have a value of 16, 9, or 4. The expansion of the vectors $\{X_{i-1}\}$ into the vectors $\{P'_i\}$ is carried out by the use of two linear filters (Fig. 5.23) and some threshold devices (Table 5.8 or 5.9). The detector then evaluates a quantity C'_i (eqn. 5.56) for each of the expanded vectors $\{P'_i\}$ here, and takes the detected data

symbol s'_{i-n} to have the value of x_{i-n} in the vector P'_i associated with the smallest value of C'_i. The m vectors $\{P'_i\}$ associated with the smallest $\{C'_i\}$ are then selected to give the m stored vectors $\{X_i\}$ having the appropriate components of the corresponding $\{P'_i\}$ here.

System 8 (Version c) : This version is the same as version b of system 8 except that the taps associated with $y_0, y_1, \cdots, y_{f-1}$ in the two linear filters (Fig. 5.23) used in version b are now removed to give the corresponding two linear filters (Fig. 5.24) for version c here, where y_f is the first significant component of the channel sampled impulse response.

It can be seen from Figs. 5.25 - 5.31 that, system 8 with version c appears to have the same performance (over the six telephone circuits tested) as system 8 with version b. System 8 with version c is therefore more attractive than the system with version b, since it requires fewer operations in the detection process.

When m = 16 and m' = 16, system 8 with version b or c is seen to have a better tolerance to noise than that of the optimum nonlinear equalizer over all the six channels, at error rates below 10^{-2}. The gains over the equalizer, at an error rate of 10^{-4}, are about 2 dB on channel A, 4.5 dB on channel B, and 1 dB on channels C, D, E, and F. Its performance over channel C also appears to be the same as that of system 1. At an error rate of 10^{-4}, the reductions in tolerance to noise of system 8 with version b or c over system 1 are about 0.5 dB on channels B, D, E, F, and 1 dB on channel A, the differences being becoming smaller as the error rates decrease further.

When m' is reduced to 9 with m = 16, the performance of system 8 with version b or c remains unchanged for all the channels tested here except for channels D and F where the system becomes inferior to the optimum nonlinear equalizer. When m' is further reduced to 4 with the same value of m, the performance of system 8 with version b or c operating over any of the channels A, C, D, E, and F is further degraded.

When m is reduced from 16 to 8 for a given value of m', the performance of system 8 with version b or c is degraded for any of the six channels tested here. Nevertheless, the system is still superior to the optimum nonlinear equalizer at higher signal to noise ratios, so long as a sufficiently large value of m' is used in the system. In particular, when m = 8 and m' = 16, the gains in the tolerance to noise over the equalizer, at an error rate of 10^{-4}, are about 1 dB on channels A and F, 3 dB on channel B, 0.5 dB on channels C and D, and 0 dB on channel E. The effect of the values of m' on the performance of system 8 with version b or c with m = 8, is roughly the same as that of the system with m = 16. That is, if the performance remains unchanged for m' = 16 and 9, with m = 16, then it will also remain unchanged for m' = 16 and 9, with m = 8.

It can be seen from the results given above and the descriptions of the channels given in section 2.5 or 5.11 that, the worse the channel, in terms of the severity of its attenuation distortion, the more improvement may be gained by the system 8 here over the optimum nonlinear equalizer, for the six channels tested here.

All the simulation results here suggest that, system 8 with version b or c is able to achieve a very satisfactory performance with the use of a moderate or small value of m (number of stored vectors X_i), especially in applications where the error rates are required

to be very low. Furthermore, the tests here have also shown that, all the m n-component stored vectors obtained at the end of the detection process here, generally, have exactly the same values for the corresponding components except for their last few components. This indicates that the value of n to be used in the detection process here could be considerably less than 16, for the given values of m considered here, without noticeably degrading the performance of the detection process, hence leading to fewer operations required in the system here.

The main advantage of system 8 with version b or c over system 1 and the optimum nonlinear equalizer is that, it can operate directly over the received sample values at the channel output (Fig. 5.23), without requiring the sampled impulse response of the channel to be adjusted into any particular form by means of a linear prefilter. This means that in adjusting the receiver adaptively for a time-varying channel, the simpler or more effective techniques avaliable to adaptive detectors, such as the direct estimation of the channel sampled impulse response using the channel estimator[A9], and the use of this estimate to achieve time synchronisation[C38] may now be used in system 8 here. One possible drawback of system 8 with version b or c is that, the additional signal processing involved in the detection process here may sometimes greatly exceed the number of operations involved in the prefilter of system 1, especially when large values of m and m' are required to be used in the system here.

Overall, system 8 with version b or c appears to be very promising for use over a time-varying channel which introduces severe attenuation distortion when the optimum nonlinear equalizer is unable to operate satisfactorily.

CHAPTER 6

SYSTEMS WITH CODED SIGNALS

6.1 Introduction

It is well known that, when a transmitted binary signal is not significantly bandlimited by the transmission path, a useful improvement in tolerance to additive white Gaussian noise can be achieved through the use of a binary convolutional code which, for a given information rate (binary data bit rate), appropriately increases the transmitted binary-digit rate. The received signal is here decoded by a maximum likelihood decoder using the Viterbi algorithm or else by a near-maximum likelihood decoder which is a suitable development of the Viterbi algorithm that is simpler to implement.(D15,D16,D29,D32,D34,D38,D40,D41,D42)

It has been recently shown that the Viterbi algorithm may also be employed for the maximum likelihood detection of an uncoded digital signal, that has been distorted in transmission over a linear channel through being convolved with the impulse response of the channel. Appropriate modifications or developments of the Viterbi algorithm, which are simpler to implement, may again be used for the near-maximum likelihood detection of the received signal.(C37,C38,C42)

More recently, tests have shown that, the Viterbi algorithm (and hence at least some of its developments) may be used for the maximum (or near-maximum) likelihood detection of a convolutionally coded digital signal that has itself been distorted through being convolved with the impulse response of the channel.(D39,D40) The detector here operates directly on the received (noisy and distorted) coded signal to give

the detected values of the original uncoded data signal at the transmitter without involving a separate decoding process. The problem with this arrangement is that the coded signal is related to the corresponding uncoded signal by having the same signal alphabet (usually binary) but having an appropriate higher symbol rate to accomodate the additional symbols introduced by the convolutional code. This means that when the symbol rate of the uncoded signal is close to the Nyquist rate for the channel, no very useful advantage in tolerance to noise is achieved through the convolutional coding of this signal because of the consequent large increase in distortion of the received signal.

It has been known for sometime that, if the very best available tolerance to additive white Gaussian noise is to be achieved at high signal to noise ratios, it is necessary to transmit a suitably coded multi-level signal, the optimum number of levels increasing with the signal to noise ratio.(A4)

A convenient arrangement here, which combines the advantages of all the facts mentioned above, is to use a binary convolutional code to convert the original binary data signal into a suitable multi-level coded signal, having the same symbol (signal-element) rate as the original uncoded signal. The received coded signal has again been distorted by the channel and is now detected by a suitable development of the Viterbi-algorithm detector, without involving a separate decoding process. Since there is here no change in the symbol rate, as a result of the coding operation, the distortion in the received multi-level coded signal is, of course, the same as that in the corresponding received uncoded signal.

The arrangement just described is studied in chapter 7 for the case when the uncoded signal is binary, and in chapter 8 for the case when the uncoded signal is a 16-point QAM signal.

6.2 Model of the Data-Transmission System for Coded Signals

The baseband model of the data-transmission system to be studied in chapters 7 and 8 is shown in Fig. 6.1. The information to be transmitted is here carried by a sequence of binary data digits $\{s_i\}$, which for $i > 0$ are statistically independent and equally likely to have any of the two values 0 and 1. When $i \leq 0$, $s_i = 0$. The coder in Fig. 6.1 converts the sequence $\{s_i\}$ into the corresponding sequence of multi-level coded symbols $\{q_i\}$. Again, $q_i = 0$ when $i \leq 0$. The conversion of $\{s_i\}$ to $\{q_i\}$ here is carried out by using a binary convolutional encoder with error correction capability to code the $\{s_i\}$ into a longer binary sequence which in turn is then mapped into the multi-level coded symbols $\{q_i\}$ by the use of a suitable Gray coder. The characteristics of the linear baseband channel in Fig. 6.1 is the same as that described in section 2.1 and will not be discussed again here. Thus, the signal at the output of the receiver filter which forms a part of the linear baseband channel is

$$r(t) = \sum_i q_i y(t - iT) + w(t) \qquad (6.1)$$

where $y(t)$ is the impulse response of the equivalent baseband channel, and $w(t)$ is the additive white Gaussian noise waveform added at the output of the receiver filter here. The waveform $r(t)$ is then sampled once per signal element at the time instants, $\{iT\}$, where i takes on all positive integer values. Let the sampled impulse response of the baseband channel here be given by the $(g+1)$-component vector

$$V = \begin{bmatrix} y_0 & y_1 & \cdots & y_g \end{bmatrix} \qquad (6.2)$$

where $y_i = y(iT)$. The delay in transmission, other than that involved

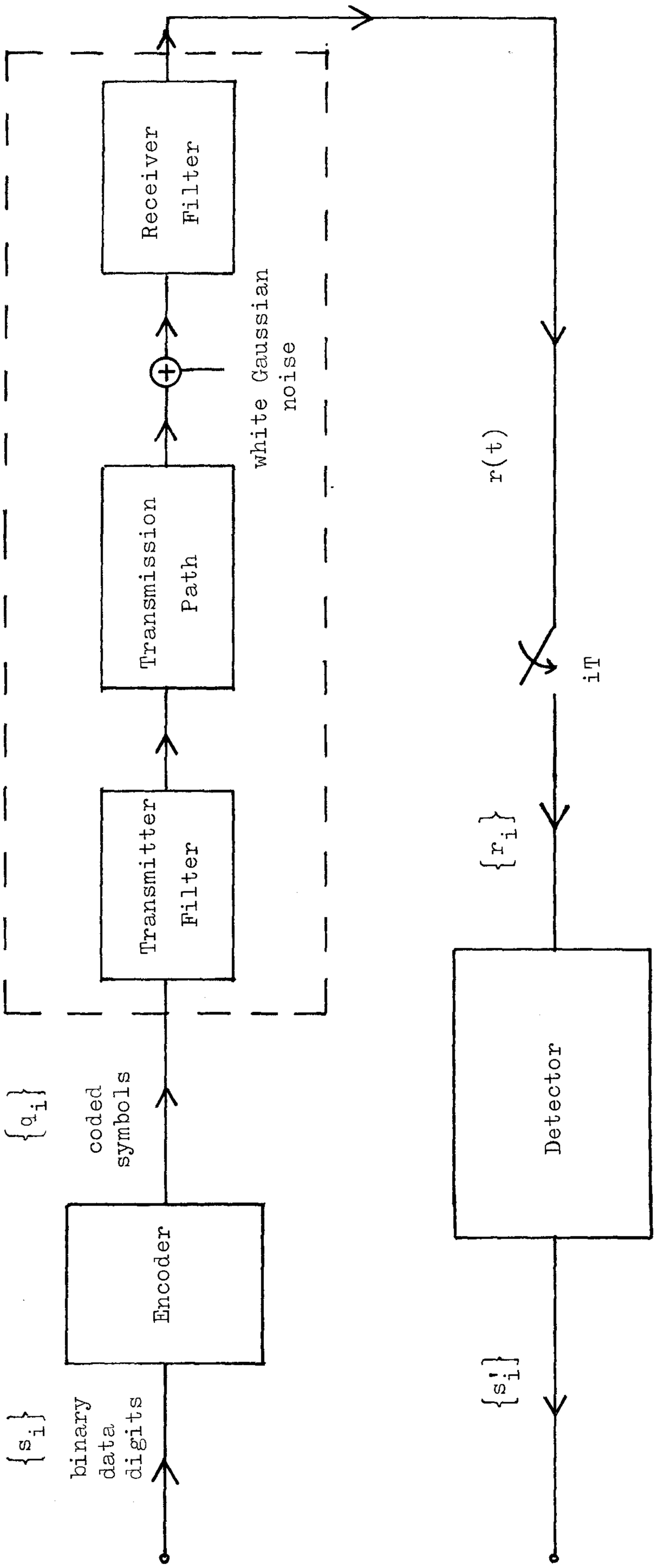


Fig. 6.1 Model of the data-transmission system for coded signals.

in the time dispersion of the transmitted signal, is neglected here so that $y_0 \neq 0$ and $y_i = 0$ for $i < 0$ and $i > g$. The sample value of the received signal at the output of the baseband channel, at time $t = iT$, is now given by

$$r_i = \sum_{h=0}^{g} q_{i-h} y_h + w_i \qquad (6.3)$$

where $r_i = r(iT)$ and $w_i = w(iT)$. The detector (Fig. 6.1) is assumed to have the prior knowledges of both V (eqn. 6.2) and the possible values of s_i and q_i. This detector employs a joint near-maximum likelihood detection and decoding process and it operates on the sample values $\{r_i\}$ to give at its output the detected values of $\{s_i\}$. The detected value of s_i is here designated as s_i'.

In chapter 8 where a QAM (Quadrature Amplitude Modulation) signal is transmitted over the channel, the values of $\{s_i\}$, $\{q_i\}$, $\{y_i\}$, $\{w_i\}$ and $\{r_i\}$ are all complex-valued quantities. The equivalent baseband model of the data-transmission system here is otherwise the same as that given above.

It is assumed that the transmitter filter (Fig. 6.1) is such that, the average transmitted energy per signal element is the mean square value of q_i, where q_i is the coded symbol carried by the ith transmitted signal element here. Furthermore, it is also assumed that, the receiver filter is such that the real and imaginary parts of the noise components $\{w_i\}$ (eqn. 6.3) are statistically independent Gaussian random variables with zero mean and a fixed variance $\sigma^2 = \frac{1}{2}N_o$, bearing in mind that the imaginary parts of $\{w_i\}$ are absent in the data-transmission system considered in chapter 7. Thus, the noise added at the output of the transmission path shown in Fig. 6.1 has a two-sided power spectral density of $\frac{1}{2}N_o$ for the binary system studied in chapter 7, or N_o for the QAM system studied in chapter 8.

6.3 Mathematical Model of a Convolutional Encoder

The input sequence $\{s_h\}$ and the output sequence $\{v'_h\}$ of a (c,b) convolutional encoder can be represented by the two vectors

$$\overline{S}_i = \begin{bmatrix} s_1 & s_2 & \cdots & s_{ib} \end{bmatrix} \tag{6.4}$$

and

$$\overline{V}_i = \begin{bmatrix} v'_1 & v'_2 & \cdots & v'_{ic} \end{bmatrix} \tag{6.5}$$

where s_h is a binary input digit (0 or 1) and v'_h is a binary output digit (0 or 1). The function of the convolutional encoder is therefore to map the vector $\overline{S}_i$ (eqn. 6.4) into the vector $\overline{V}_i$ (eqn. 6.5), and this is achieved by using a generator matrix G, where

$$\overline{V}_i = \overline{S}_i G \tag{6.6}$$

$$G = \begin{pmatrix} G_0 & G_1 & G_2 & \cdot & \cdot & G_{k-1} & 0 & 0 & \cdot & \cdot & \cdot \\ 0 & G_0 & G_1 & G_2 & \cdot & \cdot & G_{k-1} & 0 & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot & & & & & \\ & & & \cdot & \cdot & \cdot & & & & & \\ & & & \cdot & \cdot & \cdot & & & & & \end{pmatrix} \tag{6.7}$$

$$G_h = \begin{pmatrix} g_{h,1}^{(1)} & g_{h,1}^{(2)} & \cdots & g_{h,1}^{(c)} \\ g_{h,2}^{(1)} & g_{h,2}^{(2)} & \cdots & g_{h,2}^{(c)} \\ & & \cdot & \\ & & \cdot & \\ g_{h,b}^{(1)} & g_{h,b}^{(2)} & \cdots & g_{h,b}^{(c)} \end{pmatrix} \tag{6.8}$$

and $g_{h,j}^{(e)}$ has a binary value of 0 or 1, bearing in mind that a vector

is also treated as a row matrix in this thesis. For short convolutional codes, b and c are usually small integer values. Given the generator matrix G, a convolutional encoder can be constructed with bk registers and c modulo-2 adders, where b, k, and c are as defined in eqns. 6.4 - 6.8. Fig. 6.2 shows a possible arrangement of a (3,2) convolutional encoder, where the input sequence $\{s_h\}$ is divided into b (or 2 in this case) parallel streams of sequence. A symbol marked s_h is a register or store holding the binary digit s_h, and the symbol marked $\oplus$ is a modulo-2 adder. The binary digits $\{s_h\}$ in each of the b parallel streams of input sequence are here shifted 1 place to the right at each b sampling instants.

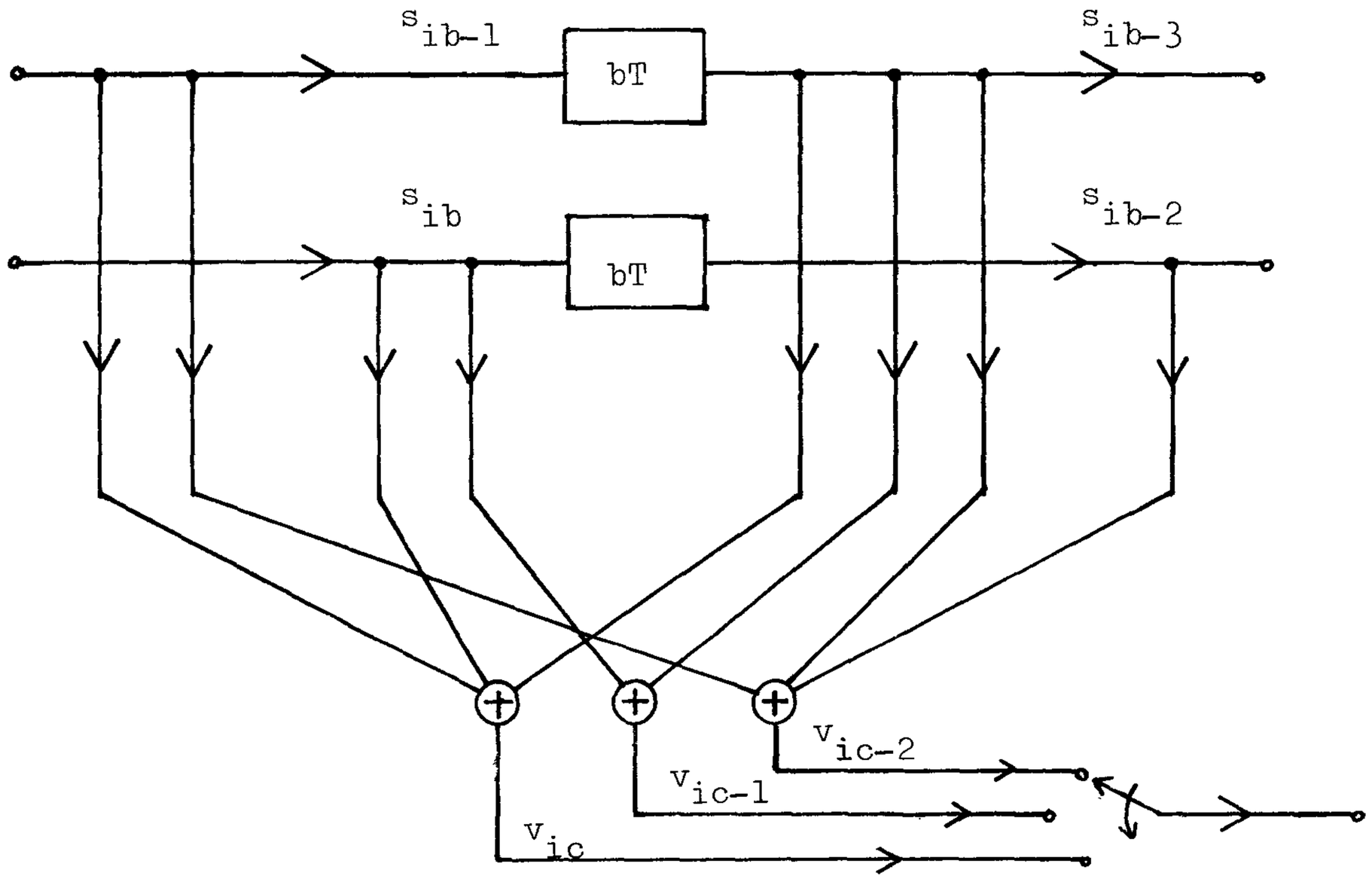


Fig. 6.2 Construction of a (c,b) convolutional encoder from its generator matrix (eqns. 6.6 - 6.8). c = 3, b = 2, k = 2.

$G_0 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $G_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$.

$\oplus$ is a modulo-2 adder.

The quantity bk is often referred to as the constraint length of the convolutional code. That is,

$$\nu = bk \quad \text{bits} \tag{6.9}$$

where ν is the constraint length of the convolutional code. Furthermore, it can be seen from eqns. 6.4 - 6.8 that, in order to produce c coded digits at the output of a (c,b) convolutional encoder (Fig. 6.2), b(k-1) previous input digits are required in the coding process here, and thus the code is said to have a memory of

$$m_e = b\,(k-1) \quad \text{bits} \tag{6.10}$$

The code rate of the encoder here is defined to be

$$R_c = b/c \quad \text{bits/code symbol} \tag{6.11}$$

One alternative arrangement [D16] of the convolutional encoder to that shown in Fig. 6.2 is to have just one (instead of b) stream of input sequence, and to use c code generators which can be derived easily from the generator matrix. Let the eth code generator be given by the ν-component vector or row matrix

$$G^{(e)} = \left[g_0^{(e)} \quad g_1^{(e)} \quad \cdots \quad g_{\nu-1}^{(e)} \right] \tag{6.12}$$

for $e = 1, 2, \cdots, c$ where ν (eqn. 6.9) is the constraint length of the convolutional code here. These c code generators are related to the generator matrix by

$$G^{(e)} = \left[g_{0,b}^{(e)} \; g_{0,b-1}^{(e)} \; \cdots \; g_{0,1}^{(e)} \; g_{1,b}^{(e)} \; \cdots \; g_{1,1}^{(e)} \; \cdots\cdots \; g_{k-1,b}^{(e)} \; g_{k-1,b-1}^{(e)} \; \cdots \; g_{k-1,1}^{(e)} \right] \tag{6.13}$$

for $e = 1, 2, \cdots, c$ where the $\left\{ g_{h,j}^{(e)} \right\}$ are as defined in eqns. 6.7 and 6.8. Fig. 6.3 shows the alternative arrangement of the convolutional encoder just described for the same example given in Fig. 6.2. The binary digits $\left\{ s_h \right\}$ here are now shifted b places at a time to the right with an interval between each shifting of b sampling instants. A value of 0 for $g_h^{(e)}$ means an open circuit, and a value of 1 for $g_h^{(e)}$ means a closed circuit.

The arrangement just described for the convolutional encoder (Fig. 6.3) is the model for the convolutional encoder considered in Fig. 6.1 for the investigations here.

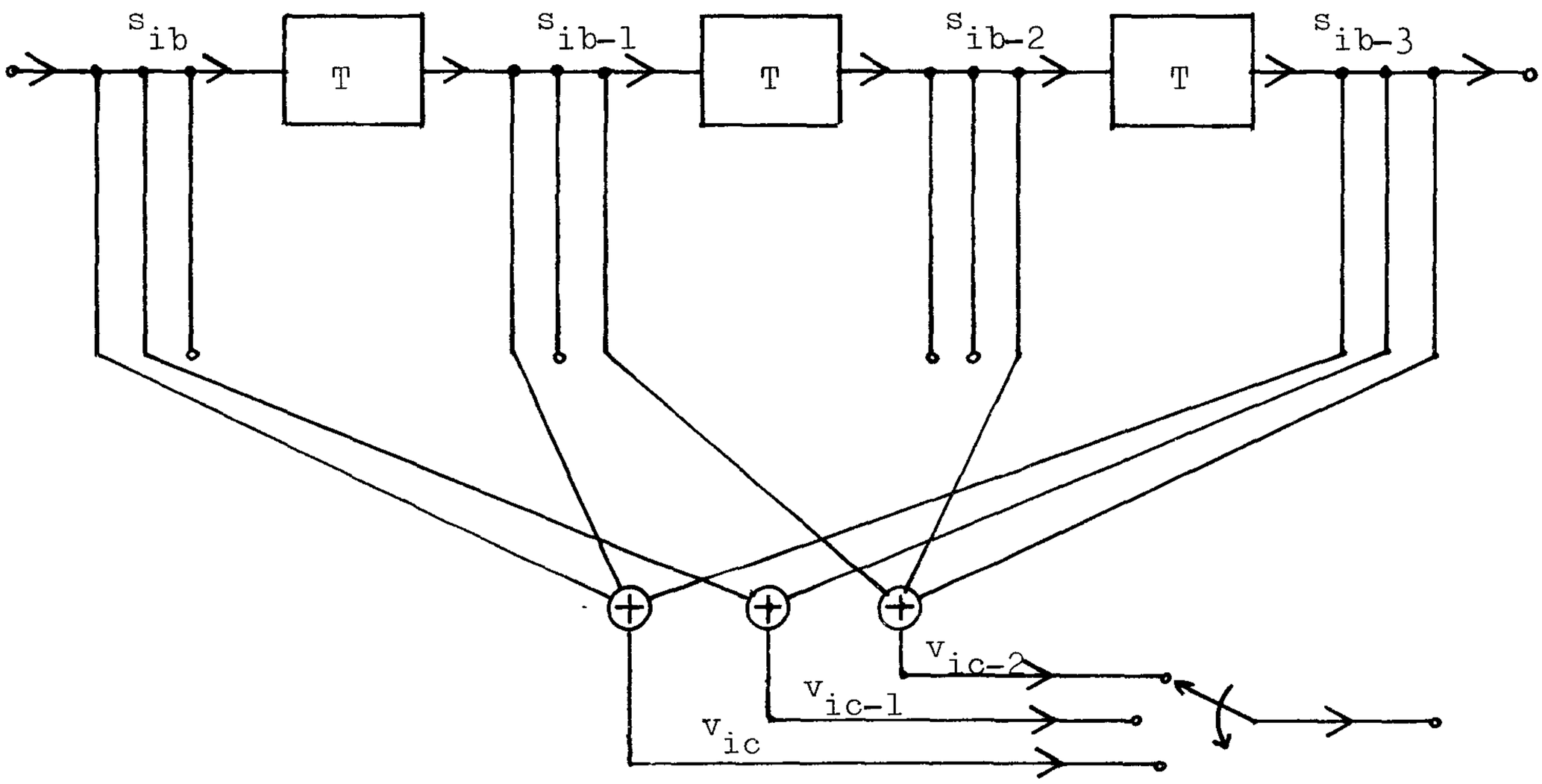


Fig. 6.3 Construction of a (c,b) convolutional encoder from its code generators (eqns. 6.12 and 6.13). c = 3, b = 2, k = 2. $G^{(1)} = \left[1\ 1\ 0\ 1 \right]$, $G^{(2)} = \left[1\ 0\ 0\ 1 \right]$, $G^{(3)} = \left[0\ 1\ 1\ 1 \right]$. $\oplus$ is a modulo-2 adder.

6.4 Viterbi-Algorithm Decoder

The Viterbi decoding algorithm is a sequential decoding algorithm used to decode the convolutionally coded sequence. It has been known that, if the convolutionally coded sequence is corrupted by stationary additive white Gaussian noise, then the Viterbi decoding can achieve, for practical purposes, the maximum likelihood decoding which minimizes the probability of error in the decoding of the whole data sequence when the binary digits in the uncoded sequence are statistically independent and equally likely to have any of their possible values.(A10) The operation of the Viterbi-algorithm docoder is described as below, assuming the communication system shown in Fig. 6.4.

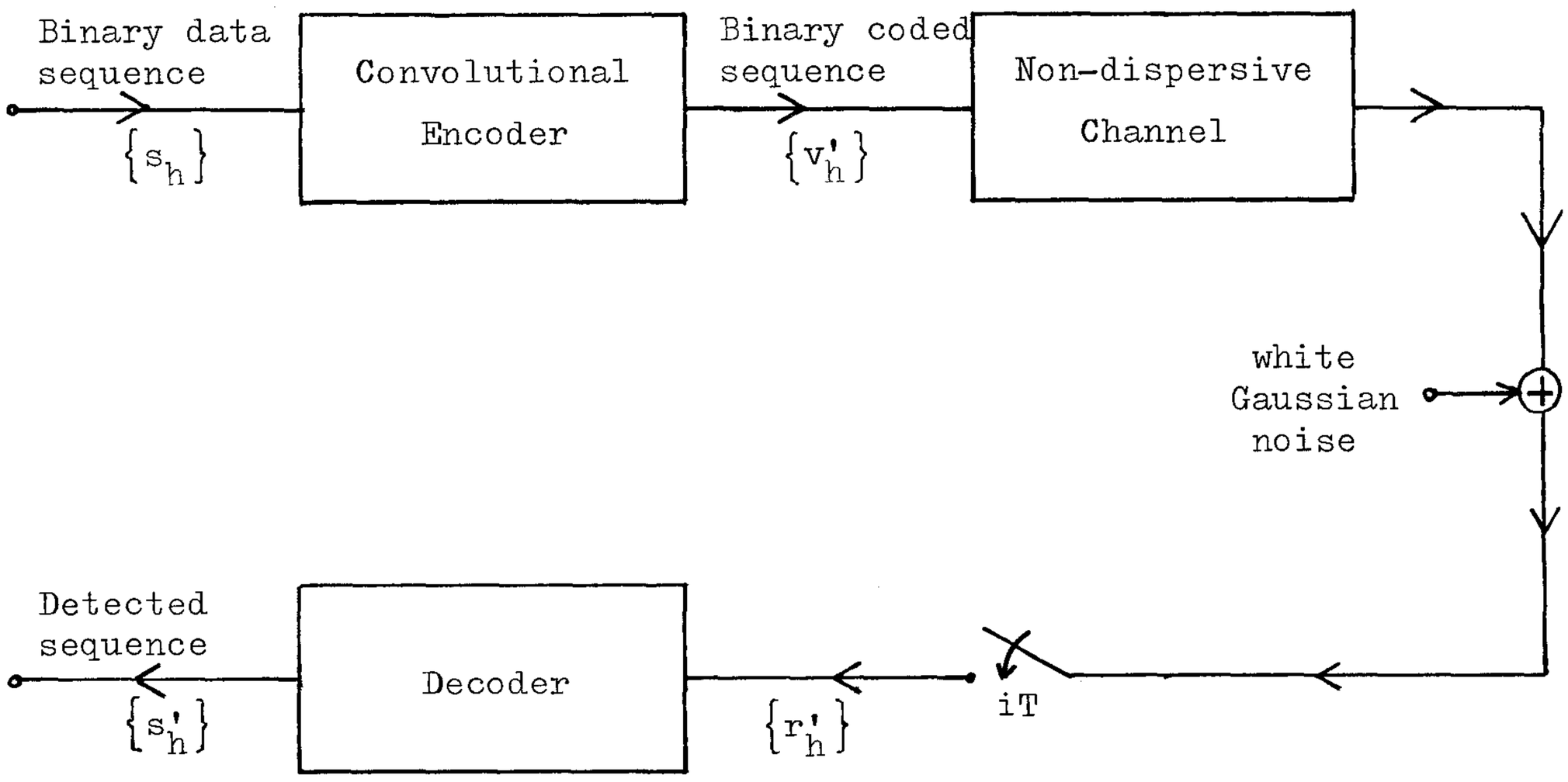


Fig. 6.4 A communication system employing a convolutional encoder over a non-dispersive channel with a stationary additive white Gaussian noise.

In Fig. 6.4, a (c,b) convolutional code is used for the encoder to convert the binary data sequence $\{s_h\}$ (eqn. 6.4) into the binary coded sequence $\{v'_h\}$ (eqn. 6.5) for transmission. A stationary white Gaussian noise is then added to the transmitted sequence to give, at the decoder input, the sample values

$$r'_h = v'_h + w'_h \tag{6.14}$$

for $h = 1, 2, \cdots, ic$ where $\{w'_h\}$ are the noise components which are the statistically independent Gaussian random variables with zero mean and a fixed variance. The decoder then operates on these sample values (eqn. 6.14) to give the decoded values for the data sequence $\{s_h\}$. For an ideal Viterbi-algorithm decoder, this is carried out as follows. Just prior to the receipts of the sample values $r'_{ic+1}, r'_{ic+2}, \cdots, r'_{ic+c}$, the Viterbi decoder holds in store m_v (ib)-component vectors

$$\overline{X}_i = \begin{bmatrix} x_1 & x_2 & \cdots & x_{ib} \end{bmatrix} \tag{6.15}$$

where x_h may have any of the two possible values of s_h (eqn. 6.4). These m_v stored vectors $\{\overline{X}_i\}$ are such that, they have all the possible combinations for their last m_e (eqn. 6.10) components, where m_e is the memory of the (c,b) convolutional code used to convert $\{s_h\}$ to $\{v'_h\}$ here. Since each component of $\overline{X}_i$ here can have a value of either 0 or 1, there are altogether $2^{(m_e)}$ possible combinations for the last m_e components of $\overline{X}_i$ here and hence

$$m_v = 2^{(m_e)} \tag{6.16}$$

Associated with each stored vector $\overline{X}_i$ is stored the corresponding value

of $\overline{C}_i$, where $\overline{C}_i$ is here defined as

$$\overline{C}_i = \overline{c}_1 + \overline{c}_2 + \cdots + \overline{c}_i \qquad (6.17)$$

and $$\overline{c}_h = \sum_{j=0}^{c-1} (r'_{hc-j} - u'_{hc-j})^2 \qquad (6.18)$$

for $h = 1, 2, \cdots, i$ where $\{u'_j\}$ are the possible values of $\{v'_j\}$ (eqn. 6.14) obtained by coding the corresponding $\{x_j\}$ (eqn. 6.15) using the same (c,b) convolutional code as that used to convert the $\{s_j\}$ into the $\{v'_j\}$. The true maximum likelihood vector $\overline{X}_i$ is the one of the stored vectors $\{\overline{X}_i\}$ for which the value of $\overline{C}_i$ is the minimum.(D42) On the receipts of the sample values $r'_{ic+1}, r'_{ic+2}, \cdots, r'_{ic+c}$, the Viterbi decoder expands each of the m_v stored vectors $\{\overline{X}_i\}$ into 2^b (ib+b)-component vectors $\{\overline{X}_{i+1}\}$ having the 2^b possible combinations for the last b components $x_{ib+1}, x_{ib+2}, \cdots, x_{ib+b}$. Each of these $(m_v)(2^b)$ expanded vectors has the associated value of

$$\overline{C}_{i+1} = \overline{C}_i + \overline{c}_{i+1} \qquad (6.19)$$

which is determined using the appropriate stored value $\overline{C}_i$. For each of the m_v (eqn. 6.16) possible combinations of the values of the last m_e (eqn. 6.10) components of the $\{\overline{X}_{i+1}\}$ (eqn. 6.15), the Viterbi decoder now selects the vector $\overline{X}_{i+1}$ associated with the smallest value of $\overline{C}_{i+1}$. The resultant m_v vectors $\{\overline{X}_{i+1}\}$ are then stored together with the associated values $\{\overline{C}_{i+1}\}$. One of these vectors is the true maximum-likelihood vector $\overline{X}_{i+1}$. The process continues in this way until the whole message has been received when all the data digits $\{s_h\}$ are decoded simultaneously to have the values of the corresponding components $\{x_h\}$ of the true maximum likelihood vector.

The decoding process just described is the ideal arrangement of the Viterbi decoding algorithm which achieves the maximum likelihood decoding. In practice, the data digits $\{s_h\}$ are decoded at a delay before the receipt of the whole message so as to reduce the number of stored digits required in the decoder. Thus, the b data digits $s_{(i-n)b}, s_{(i-n)b-1}, \cdots, s_{(i-n)b-(b-1)}$ are now decoded to have the values of the corresponding components $x_{(i-n)b}, x_{(i-n)b-1}, \cdots, x_{(i-n)b-(b-1)}$ in the true maximum likelihood vector $\overline{X}_i$ (that associated with the smallest value of $\overline{C}_i$). The integer n is preferably much larger than the memory m_e (eqn. 6.10) of the convolutional code. After the decoding of $s_{(i-n)b}, s_{(i-n)b-1}, \cdots s_{(i-n)b-(b-1)}$, the decoder does not need to consider the values of $x_{(i-n)b}, x_{(i-n)b-1}, \cdots\cdots$ in the following decoding processes. Thus, instead of storing m_v (ib)-component vectors $\{\overline{X}_i\}$, the decoder here stores the corresponding m_v (nb)-component vectors $\{X_i\}$, where

$$X_i = \left[x_{(i-n)b+1} \quad x_{(i-n)b+2} \quad \cdots \quad x_{ib} \right] \tag{6.20}$$

so that X_i is formed by the last nb components of the corresponding vector $\overline{X}_i$.

The drawback of the Viterbi-algorithm decoder is that the number of operations and the amount of storage involved in the decoding process here increases exponentially with the memory m_e of the convolutional code. Consequently, when a large value of m_e is used, the decoding process here may become unduly complex, and a near-maximum likelihood decoding process[D42] that involves fewer number of operations and less amount of storage must now be used instead.

6.5 Optimum Free Distance Rate $\frac{1}{2}$ and $\frac{2}{3}$ Convolutional Codes

The free distance d_∞ of a convolutional code is defined to be the minimum Hamming distance between some coded sequences $\{\overline{V}_\infty\}$ (eqn. 6.5) generated from a data sequence (eqn. 6.4) with $s_1 = 1$, and some $\{\overline{V}_\infty\}$ with $s_1 = 0$. d_∞ is also the minimum Hamming weight of the coded sequence $\overline{V}_\infty$ generated from the data sequence with $s_1 = 1$. The significance of the free distance d_∞ is that it appears to be the principal determiner of the probability of error in decoding a convolutionally coded sequence with the maximum likelihood (or nearly so) decoding process.(D30) This being such that, the larger the value of d_∞, the lower may be the probability of error in the decoding process.

Some optimum (maximal) free distance rate $\frac{1}{2}$ and $\frac{2}{3}$ nonsystematic convolutional codes have been found by Johannesson(D30) and Paaske(D27). The rate $\frac{1}{2}$ codes that are used in this thesis are those found by Johannesson, whereas the rate $\frac{2}{3}$ codes that are used in this thesis are those found by Paaske. Tables 6.1 and 6.2 give the corresponding code generators for these codes. It can be seen from Tables 6.1 and 6.2 that, a code having a larger constraint length ν (eqn. 6.9) also has a larger optimum free distance. Furthermore, for a given value of ν, the optimum free distance of a rate $\frac{1}{2}$ code appears to be larger than that of the corresponding rate $\frac{2}{3}$ code. One drawback of using a code with a larger constraint length is that, the equipment complexity involved in implementing the encoder and decoder here also increases. The values of the constraint length are therefore selected to be not more than 10 for the investigations here.

Code memory m_e	Constraint length ν	Code generators $G^{(1)}$	Code generators $G^{(2)}$	Free distance d_∞
3	4	1111	1011	6
5	6	111101	101101	8
7	8	11001011	10111101	10
10	11	11110100101	10110111001	14

Table 6.1 Code generators (eqns 6.12 and 6.13) for the optimum rate $\frac{1}{2}$ binary convolutional codes.(D30)

Code memory m_e	Constraint length ν	Code generators $G^{(1)}$	Code generators $G^{(2)}$	Code generators $G^{(3)}$	Free distance d_∞
4	6	011101	100011	111010	5
7	9	0101101010	1001011110	1110111100	8
10	12	011110001101	101001110011	111000010110	10

Table 6.2 Code generators (eqns. 6.12 and 6.13) for the optimum rate $\frac{2}{3}$ binary convolutional codes.(D27)

6.6 Channel Sampled Impulse Responses to be used for the Computer Simulation Tests in Chapters 7 and 8

These channels are used for the computer simulation tests to determine the tolerance to Gaussian noise of the various systems developed in chapters 7 and 8, assuming the data-transmission system of Fig. 6.1. Table 6.3 gives the three channel sampled impulse responses for chapter 7 where the binary data sequence is coded into a quaternary coded sequence for transmission. In Table 6.3, channel A is a non-dispersive channel which does not introduce any signal distortion. Channel B introduces a typical combination of amplitude and phase distortions, and channel C introduces considerable bandlimiting of the signal leading to severe amplitude distortion.

For the QAM system studied in chapter 8, four of the six sampled impulse responses given in Tables 2.6 and 2.7 are considered for the computer simulation tests here. These sampled impulse responses have the same property that all the zeros (or roots) of their z-transform lie inside the unit circle in the z-plane. Table 6.4 gives the values of these sampled impulse responses. In Table 6.4, channels A and B introduce very severe attenuation distortion, whereas channels C and D introduce negligible level of distortion.

The sampled impulse response of each of the channels shown in Tables 6.3 and 6.4 is normalised, so that

$$\sum_{h=0}^{g} |y_h|^2 = 1 \qquad (6.21)$$

where $y_0, y_1, \cdots, y_g$ are the g+1 components of the channel sampled impulse response, and $|y_j|$ is the modulus or absolute value of y_j.

Channels	Sampled Impulse Responses				
A	1.000	0.000	0.000	0.000	0.000
B	0.548	0.789	0.273	-0.044	0.012
C	0.167	0.471	0.707	0.471	0.167

Table 6.3 Channel sampled impulse responses to be used in chapter 7.

Channel A		Channel B		Channel C		Channel D	
Real part	Imaginary part	Real part	Imaginary part	Real part	Imaginary part	Real part	Imaginary part
0.4211	-.4187	-.1998	-.2576	0.9335	-.1289	0.8702	-.0360
0.6547	0.2704	0.4609	-.4590	0.3278	0.0183	0.4450	0.1566
-.2270	0.2622	0.2913	0.4834	-.1294	-.0182	-.1262	-.0020
-.0062	-.1387	-.3389	-.0154	0.0270	-.0006	0.0257	-.0095
0.0292	0.0441	0.1233	-.1104	0.0020	0.0011	0.0084	0.0064
-.0090	-.0073	-.0143	0.0587	-.0026	-.0013	-.0120	-.0029
0.0026	0.0044	0.0040	-.0153	-.0012	0.0008	0.0089	0.0002
-.0053	-.0010	-.0062	0.0117	-.0011	-.0003	-.0095	-.0008
0.0057	-.0011	0.0016	-.0078	0.0018	0.0007	0.0073	0.0018
-.0031	0.0001	0.0026	0.0040	-.0012	0.0002	-.0039	-.0016
0.0006	0.0006	-.0007	0.0007	-.0009	-.0003	0.0002	0.0021
-.0007	-.0040	-.0014	0.0002	0.0015	-.0003	0.0011	-.0019
0.0007	0.0015	0.0001	0.0000	-.0005	0.0001	-.0003	0.0017
-.0002	-.0021	-.0014	0.0022	-.0006	0.0003	-.0007	-.0007
-.0005	-.0002	-.0001	-.0026	0.0006	-.0002	0.0001	-.0002
0.0013	0.0002	0.0020	-.0002	0.0000	0.0004	0.0002	0.0005
-.0012	0.0001	0.0004	0.0022	0.0002	-.0004	-.0001	-.0004
0.0002	0.0001	-.0014	-.0012	-.0004	0.0000	-.0003	-.0001
0.0000	0.0001	0.0012	-.0004	0.0000	0.0002	-.0003	0.0003
0.0002	-.0001	0.0003	0.0001	0.0004	-.0002	0.0004	-.0001
0.0001	0.0008	0.0002	0.0003	-.0001	0.0002	0.0001	0.0001
-.0005	0.0003	-.0007	0.0000	-.0002	-.0001	-.0003	-.0001
0.0001	-.0001	-.0002	-.0008	0.0002	0.0001	0.0002	0.0000
0.0004	0.0005	0.0006	-.0003	0.0003	-.0001	0.0003	0.0001
-.0002	0.0001	0.0000	0.0004	-.0004	-.0001	-.0003	-.0002
0.0001	-.0001	-.0002	-.0001	0.0000	0.0002	0.0000	0.0001
0.0000	-.0004	0.0001	-.0001	0.0002	-.0001	0.0004	-.0001
0.0002	-.0001	0.0000	0.0000	0.0000	0.0001	-.0001	0.0002
-.0003	0.0000	0.0000	0.0000	-.0002	0.0000	-.0001	-.0001
0.0000	-.0004	0.0000	0.0000	0.0001	0.0000	0.0001	0.0001
0.0003	-.0001			0.0001	0.0000	0.0002	0.0000
-.0001	0.0001			-.0002	-.0001	-.0002	-.0001
-.0002	-.0001			0.0001	0.0001	0.0001	0.0001
-.0001	0.0000			0.0000	-.0001	0.0000	0.0000
0.0001	-.0001			0.0001	0.0000	0.0001	-.0001
0.0002	0.0002			0.0000	0.0000	0.0000	0.0001
-.0001	0.0003			-.0002	0.0000	-.0002	0.0000
0.0000	0.0000			0.0002	0.0000	0.0002	0.0000
0.0001	0.0001			0.0001	0.0000	0.0000	0.0000
0.0000	0.0001			-.0001	0.0000	-.0001	0.0000
0.0000	-.0001			0.0001	0.0000	0.0000	0.0000
0.0001	0.0000			0.0000	0.0000	0.0000	0.0000
0.0000	0.0001			-.0001	0.0000	0.0000	0.0000
0.0000	0.0000			-.0001	0.0000	0.0000	0.0000
0.0000	0.0000			0.0000	0.0000	0.0000	0.0000

Table 6.4 Channel sampled impulse responses to be used in chapter 8.

CHAPTER 7

CODED SYSTEM FOR BINARY SIGNALS

7.1 Introduction

This chapter is concerned with the study of a coded system which converts the binary data sequence into the appropriate quaternary coded sequence for transmission, assuming the data-transmission system described in section 6.2. Two different coding schemes have been developed here and are described in sections 7.3 and 7.4 respectively. The coded system studied here employs a joint near-maximum likelihood detection/decoding process which is described in section 7.5. The tolerance to Gaussian noise of the coded system is here compared with that of the uncoded system by using computer simulation tests, and the results and discussions are given in sections 7.7 and 7.8 respectively.

7.2 Basic Assumptions

The model of the data-transmission system for this chapter has been described in section 6.2. The coder in Fig. 6.1 is here implemented as a rate $\frac{1}{2}$ binary convolutional coder and a Gray coder to convert the binary data sequence $\{s_i\}$ into the appropriate quaternary coded sequence $\{q_i\}$ for transmission, hence having the same symbol (signal element) transmission rate as the original uncoded binary signals. The $\{s_i\}$ here are assumed to be statistically independent and equally likely to have any of the two values 0 and 1. The average transmitted energy per signal element is assumed to be equal to the mean square value

of the $\{q_i\}$, where q_i is the coded symbol carried by the ith transmitted signal element here. The baseband channel (Fig. 6.1) here is, in general, time dispersive, and it has a sampled impulse response of g+1 real-valued components $y_0, y_1, \cdots, y_g$ which are assumed to be known and time-invariant for the purpose of study here. Stationary white Gaussian noise is assumed to be the only noise added at the output of the transmission path shown in Fig. 6.1, to give, at the detector input, the noise sample values $\{w_i\}$ which are the statistically independent Gaussian random variables with zero mean and a fixed variance $\frac{1}{2}N_o$. The detector (Fig. 6.1) here employs a joint near-maximum likelihood detection/decoding process, and it operates on the received sample values $\{r_i\}$ (eqn. 6.3) to give at its output the detected values of the data digits $\{s_i\}$.

7.3 Coder 1

The schematic diagram of coder 1 is shown in Fig. 7.1. As Fig. 7.1 shows, this coder is implemented as a rate $\frac{1}{2}$ binary convolutional coder followed by a Gray coder. In converting the binary data digits $\{s_i\}$ into the quaternary coded symbols $\{q_i\}$, the coder here first codes the $\{s_i\}$ into a longer binary coded sequence by using the convolutional coder. This is carried out by determining, for each q_i, the two quantities

$$v_i^{(e)} = \sum_{h=0}^{\nu-1}{}' s_{i-h} g_h^{(e)} \qquad (7.1)$$

for $e = 1, 2$ where the $\{g_h^{(e)}\}$ are the components of the code generators $\{G^{(e)}\}$ (eqn. 6.12) of the convolutional code with constraint length ν, and $\sum'$ is a modulo-2 adder. Since modulo-2 addition is used in eqn. 7.1,

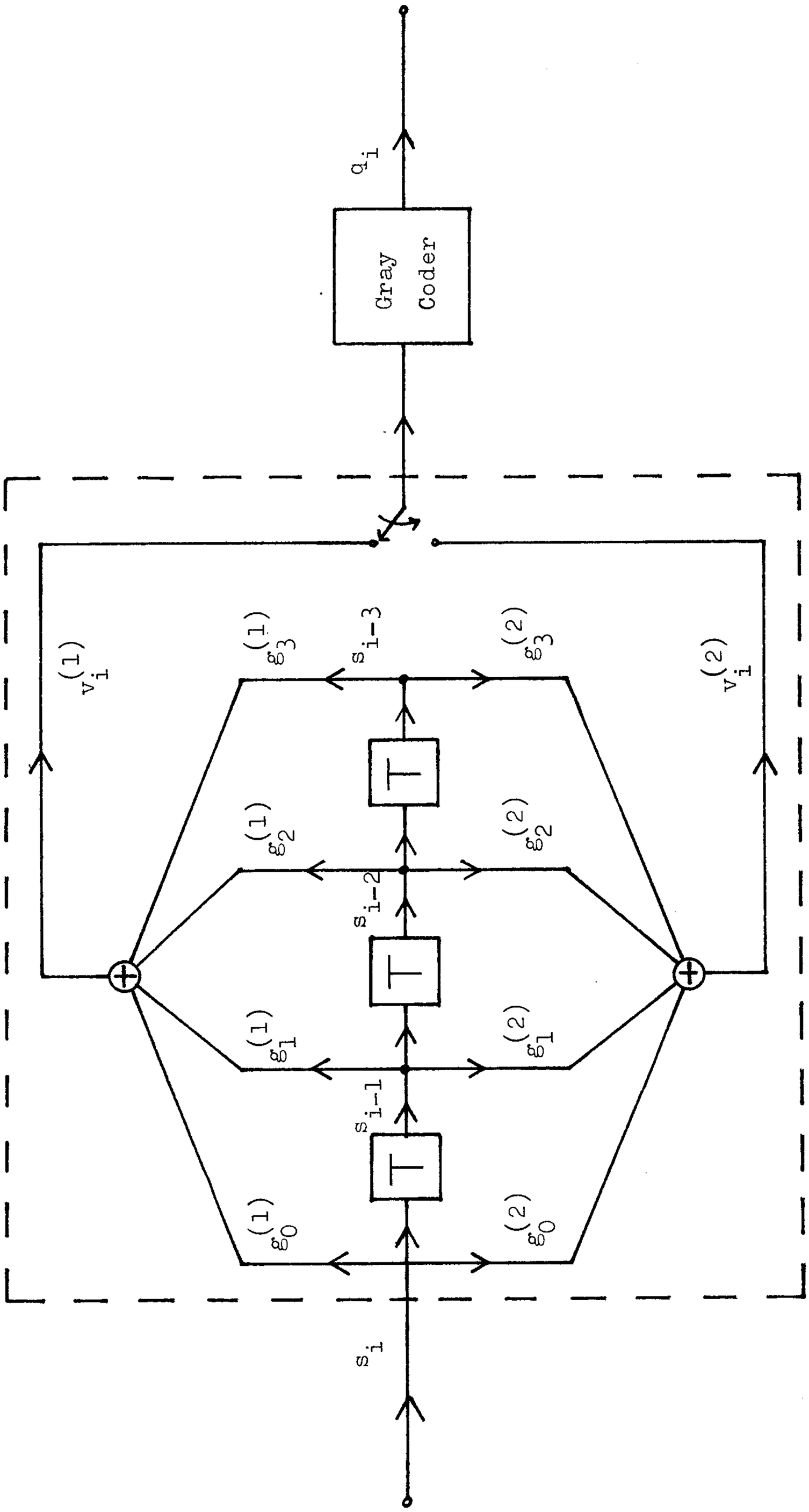


Fig. 7.1 An example of the schematic diagram of coder 1 with $r = 4$. r is the constraint length of the convolutional code having the code generators $\{g_h^{(e)}\}$. $\oplus$ is a modulo-2 adder.

the value of $v_i^{(e)}$ is either 0 or 1 so that the 2-component vector

$$V_i = \left[v_i^{(1)} \quad v_i^{(2)} \right] \tag{7.2}$$

has one of the four possible values 00, 01, 10, and 11. The value of q_i is now determined from the vector V_i using the Gray code shown in Table 7.1. A Gray code is used here, so that, for adjacent values of q_i, the corresponding vectors $\{V_i\}$ differ in only one component. Since the $\{s_i\}$ are statistically independent and equally likely to have any of their two possible values 0 and 1, it can be seen from eqn. 7.1 that, the $\{v_i^{(e)}\}$ are also statistically independent and equally likely to have any of their two possible values 0 and 1. Consequently, it can be seen from eqn. 7.2 and Table 7.1 that, the coded symbol q_i is also equally likely to have any of its four possible values. Thus, the mean-square value of $\{q_i\}$ here is equal to unity, bearing in mind that the mean-square value of $\{q_i\}$ is here assumed to be equal to the average transmitted energy per signal element. It has, in fact, been observed to be true in the computer simulation tests that over a sufficiently long sequence of sample values $\{q_i\}$, the mean-square value of these samples approaches unity.

V_i	q_i
0 0	$-3/\sqrt{5}$
0 1	$-1/\sqrt{5}$
1 1	$1/\sqrt{5}$
1 0	$3/\sqrt{5}$

Table 7.1 Relationship between the vector V_i (eqn. 7.2) and the coded symbol q_i (Fig. 7.1).

7.4 Coder 2

An effective increase in the constraint length of the convolutional code used in coder 1 may be achieved quite simply through the process of interleaving, and this leads to the development of coder 2 here. Thus, in converting the data digits $\{s_i\}$ into the coded symbols $\{q_i\}$, coder 2 first determines the two quantities

$$v_i^{(e)} = \sum_{h=0}^{\nu-1}{}' \, s_{i-hd} g_h^{(e)} \qquad (7.3)$$

for $e = 1, 2$ where d is the interleaved gap here, the $\{g_h^{(e)}\}$ are the components of the code generators of the convolutional code used in coder 1, and $\sum'$ is a modulo-2 adder. The coder here then forms the 2-component vector V_i (eqn. 7.2) and it uses Table 7.1 to determine the corresponding value for the coded symbol q_i. The value of d considered in the investigations here is 5, and the schematic diagram of an example of this coder is shown in Fig. 7.2. It should be noted that, when $d = 1$, coder 2 reduces exactly to coder 1 as can be seen from eqns. 7.1 and 7.3.

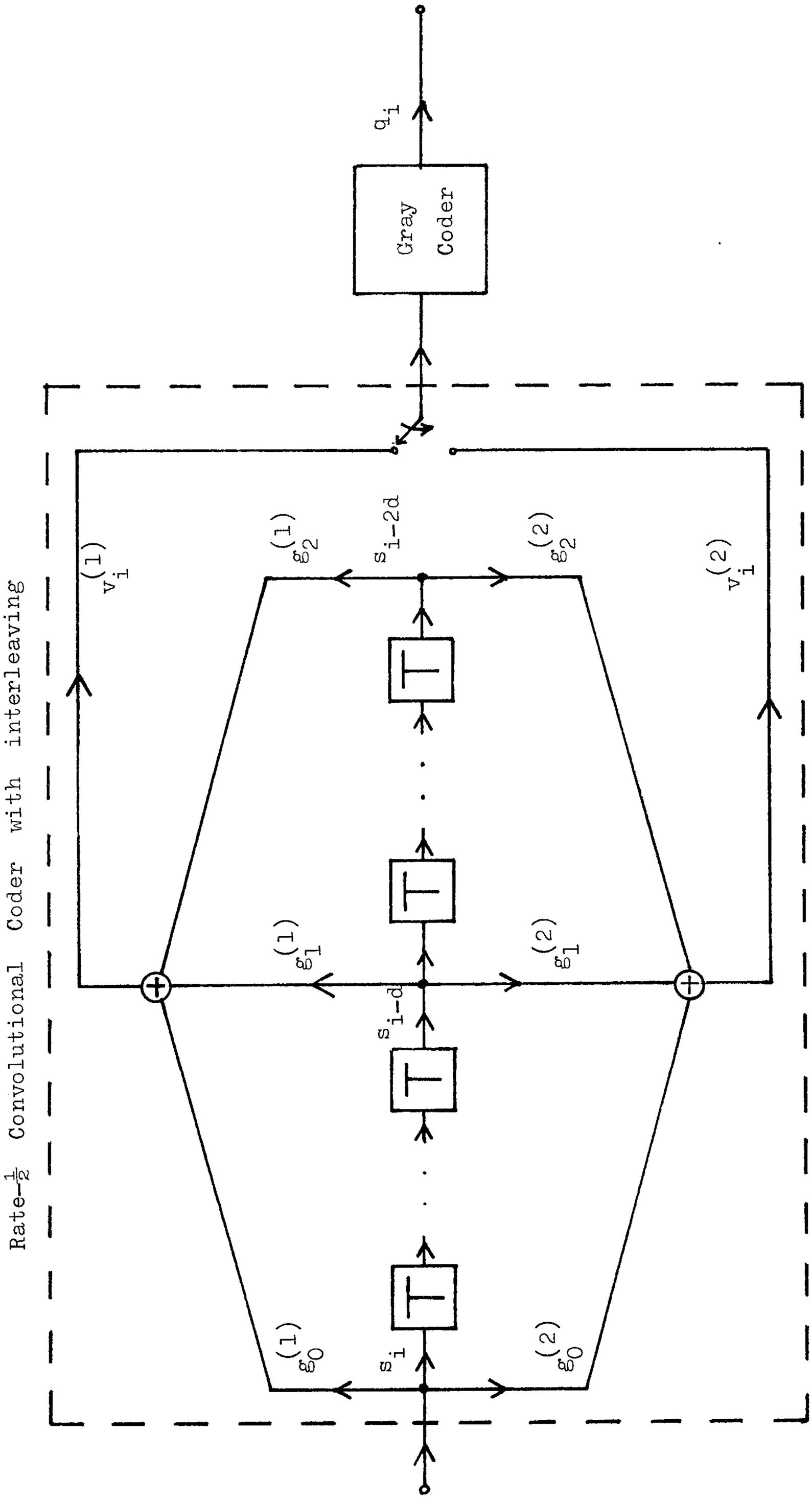


Fig. 7.2 An example of the schematic diagram of coder 2 with $\nu = 3$. $d = 5$. ν is the constraint length of the convolutional code having the code generators $\{g_h^{(e)}\} \cdot \oplus$ is a modulo-2 adder.

7.5 Detector

The detector (Fig. 6.1) here employs a joint near-maximum likelihood detection/decoding process, and it operates on the received sample values $\{r_i\}$ (eqn. 6.3) to give at its output the detected values for the data digits $\{s_i\}$. The received sample value r_i at time $t = iT$ is, from eqn. 6.3, given by

$$r_i = \sum_{h=0}^{g} q_{i-h} y_h + w_i \qquad (7.4)$$

where $\{q_i\}$ are the transmitted coded symbols, $\{w_i\}$ are the noise components, and $y_0, y_1, \cdots, y_g$ are the g+1 components of the channel sampled impulse response. The detector here operates as below.

Just prior to the receipt of r_i, the detector holds in store m n-component vectors $\{X_{i-1}\}$ where

$$X_{i-1} = \begin{bmatrix} x_{i-n} & x_{i-n+1} & \cdots & x_{i-1} \end{bmatrix} \qquad (7.5)$$

and x_h may have any of the two possible values (0 and 1) of the data digit s_h for any possible value of h. The value of n here is selected to be such that

$$n > g + (\nu' - 1)d \qquad (7.6)$$

where ν' is the constraint length of the convolutional code used at the transmitter, and d is the interleaved gap of the code. Associated with each stored vector X_{i-1} , is stored the corresponding value of cost C_{i-1} which will shortly be considered.

On the receipt of r_i, each vector X_{i-1} is expanded into two (n+1)-component vectors $\{P_i\}$, where

$$P_i = \left[\begin{array}{ccc} x_{i-n} & x_{i-n+1} & \cdots \quad x_i \end{array} \right] \qquad (7.7)$$

The first n components of P_i are as in the original vector X_{i-1}, and the last component x_i has the two possible values of s_i. Associated with each vector P_i, are the g+1 scalar quantities $\{u_j^{(1)}\}$ and the g+1 scalar quantities $\{u_j^{(2)}\}$, for $j = i, i-1, \cdots, i-g$, where

$$u_j^{(e)} = \sum_{h=0}^{\nu-1}{}' x_{j-hd} g_h^{(e)} \qquad (7.8)$$

for $e = 1, 2$, and the quantities d, ν, and $\{g_h^{(e)}\}$ are as defined in eqn. 7.3. Since modulo-2 addition is used in eqn. 7.8, the value of $u_j^{(e)}$ is either 0 or 1. These values of $\{u_j^{(e)}\}$ can be used to form the g+1 2-component vectors

$$U_j = \left[\begin{array}{cc} u_j^{(1)} & u_j^{(2)} \end{array} \right] \qquad (7.9)$$

for $j = i, i-1, \cdots, i-g$. These vectors in turn determine the g+1 scalar quantities $\{q'_j\}$, $j = i, i-1, \cdots, i-g$, where the relationship between q'_j and U_j is identical to that between q_j and V_j in Table 7.1. The g quantities $\{q'_j\}$, $j = i-1, i-2, \cdots, i-g$, have already been determined for each vector X_{i-1} in the previous detection process, and so only the quantity q'_i is required to be determined for each of the 2m expanded vectors $\{P_i\}$ here, following the receipt of r_i. The g+1 quantities $\{q'_j\}$, $j = i, i-1, \cdots, i-g$, associated with any one of the 2m vectors $\{P_i\}$, determine the scalar quantity

$$z'_i = \sum_{h=0}^{g} q'_{i-h} y_h \qquad (7.10)$$

This is the corresponding estimate of the quantity

$$z_i = \sum_{h=0}^{g} q_{i-h} y_h \qquad (7.11)$$

which is the signal component in the received sample value r_i (eqn. 7.4). Having evaluated the quantity z'_i (eqn. 7.10) for each of the 2m expanded vectors $\{P_i\}$, the detector here then evaluates the cost C_i for each of these $\{P_i\}$ as

$$C_i = C_{i-1} + (r_i - z'_i)^2 \qquad (7.12)$$

where C_{i-1} has already been determined for each vector X_{i-1} in the previous detection process. It can be seen from eqn. 7.12 that

$$C_i = \sum_{h=1}^{i} (r_h - z'_h)^2 \qquad (7.13)$$

so that the cost C_i is in fact the square of the Euclidean distance between the corresponding estimate Z'_i of the received sequence of sample values in the absence of noise, and the sequence of sample values actually received R_i, where

$$Z'_i = \begin{bmatrix} z'_1 & z'_2 & \cdots & z'_i \end{bmatrix} \qquad (7.14)$$

$$R_i = \begin{bmatrix} r_1 & r_2 & \cdots & r_i \end{bmatrix} \qquad (7.15)$$

The detector now holds 2m vectors $\{P_i\}$ together with their associated costs $\{C_i\}$. It then selects, from these vectors, a total of m vectors as follows. When m = 32, the detector selects, for each of the two

possible values of $x_{i-(m/2)+2}$, the vector P_i associated with the smallest cost C_i, to give two selected vectors having the two possible values of $x_{i-(m/2)+2}$. This process is repeated, in turn, for $x_{i-(m/2)+3}$, $x_{i-(m/2)+4}$, $\cdots$, x_i, a vector once selected not being available for selection a second time, so that no vector can be selected more than once. Having selected the m-2 vectors $\{P_i\}$ using the selection process just described, the detector next selects, from the remaining vectors, the two vectors $\{P_i\}$ associated with the smallest costs, to give the required total of m selected vectors. When m = 8, the detector begins the selection process with the selection of the two vectors $\{P_i\}$ having the two possible values of $x_{i-(m/2)+1}$ and associated with the smallest costs $\{C_i\}$, and continues in this way with $x_{i-(m/2)+2}$, $x_{i-(m/2)+3}$, $\cdots$, x_i, to give the m selected vectors $\{P_i\}$. Having selected the m vectors $\{P_i\}$, the detector then takes the detected value s'_{i-n} of the data digit s_{i-n} to have the value of x_{i-n} in the vector P_i associated with the smallest cost. The first component x_{i-n} of each of the m selected vectors $\{P_i\}$ is now discarded to give the corresponding m vectors $\{X_i\}$ which are then stored together with their assocaited costs $\{C_i\}$. The detector is now ready for the receipt of r_{i+1}, and the detection process continues in this way.

During the selection process for the m vectors $\{P_i\}$ described above, the detector may occasionally fail to find a vector available for selection. In this case, the detector selects an arbitrary vector to which it ascribes an extremely high cost value. This vector is then discarded at the next detection process.

To start the detection process here, the detector stores m vectors $\{X_n\}$ which are all the same and correct, a known synchronizing signal being transmitted here, and to one of these vectors it ascribes a cost

value of zero, whereas to each of the remainders it ascribes a very high cost value. Correct operation of the detection process is now normally achieved after the receipt of only a few sample values $\{r_i\}$.

7.6 Uncoded System

The uncoded system operates in the same way as does the coded system except that the conversion of the data digits $\{s_i\}$ to the transmitted symbols $\{q_i\}$ is here given by

$$q_i = 2\,(s_i - \tfrac{1}{2}) \qquad (7.16)$$

for $i > 0$. Thus, $q_i = -1$ when $s_i = 0$, and $q_i = 1$ when $s_i = 1$.

7.7 Computer Simulation Results

Computer simulation tests have been carried out to determine the tolerances to Gaussian noise of the coded and the uncoded systems studied in this chapter. The sampled impulse responses of the three channels A, B, and C tested here are given in Table 6.3. The results of the tests are shown in Figs. 7.3 - 7.8 where the various systems considered here are shown in Table 7.2. The 95% confidence limits of the results shown in Figs. 7.3 - 7.8 are about ± 0.5 dB and are slightly wider at the lower error rates ($< 10^{-3}$). The signal to noise ratio here is defined as

$$\text{SNR} = 10 \log_{10}\left(\frac{E'}{\frac{1}{2}N_o}\right) \quad \text{dB} \qquad (7.17)$$

where E' is the average transmitted energy per signal element, and

$\frac{1}{2}N_o$ is the two-sided power spectral density of the noise added at the output of the transmission path shown in Fig. 6.1.

All computer simulation tests have been carried out by using the CDC 7600 computer at Manchester, and the computer programs are written in FORTRAN. Appendix B3 gives the computer program for the coded system with coder 1 here.

Systems	d	m_e	m	n	Coding
A	1	0	32	16 or 64	uncoded
B1	1	7	32	64	
B2	1	5	32	64	
B3	1	3	32	64	
C1	1	7	32	16	
C2	1	5	32	16	coder 1
C3	1	3	32	16	
D1	1	7	8	16	
D2	1	5	8	16	
E1	5	7	32	128	
E2	5	5	32	128	
E3	5	3	32	128	coder 2
F1	5	7	32	64	
F2	5	5	32	64	
F3	5	3	32	64	

Table 7.2 Systems tested (Figs. 7.3 - 7.8). d is the interleaved gap and m_e is the memory of the convolutional code, m is the number of stored vectors in the detector, and n is the number of components in each stored vector.

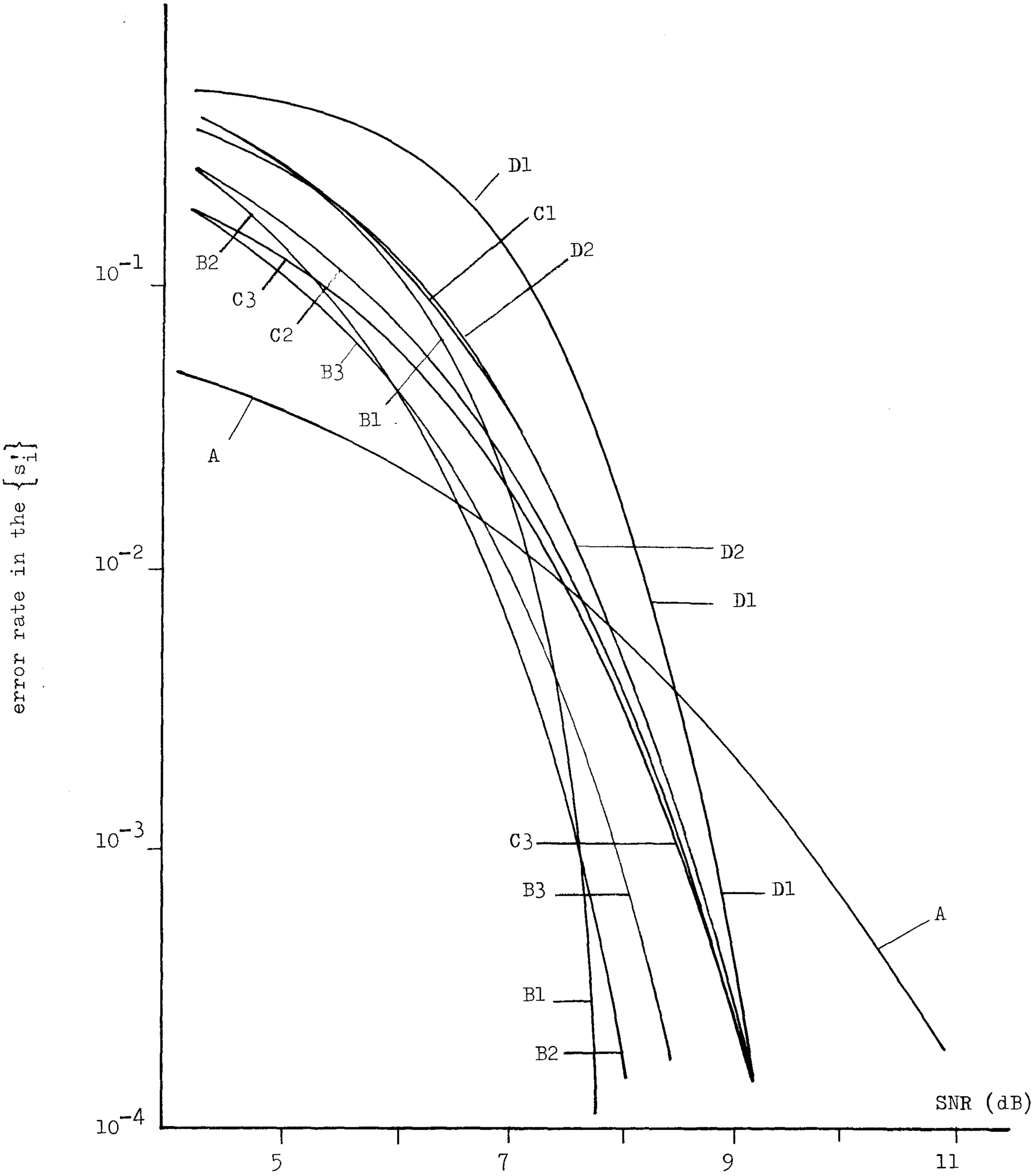


Fig. 7.3 Variation of error rate with signal to noise ratio for the various systems with coder 1 (Table 7.2) operating over channel A (Table 6.3).

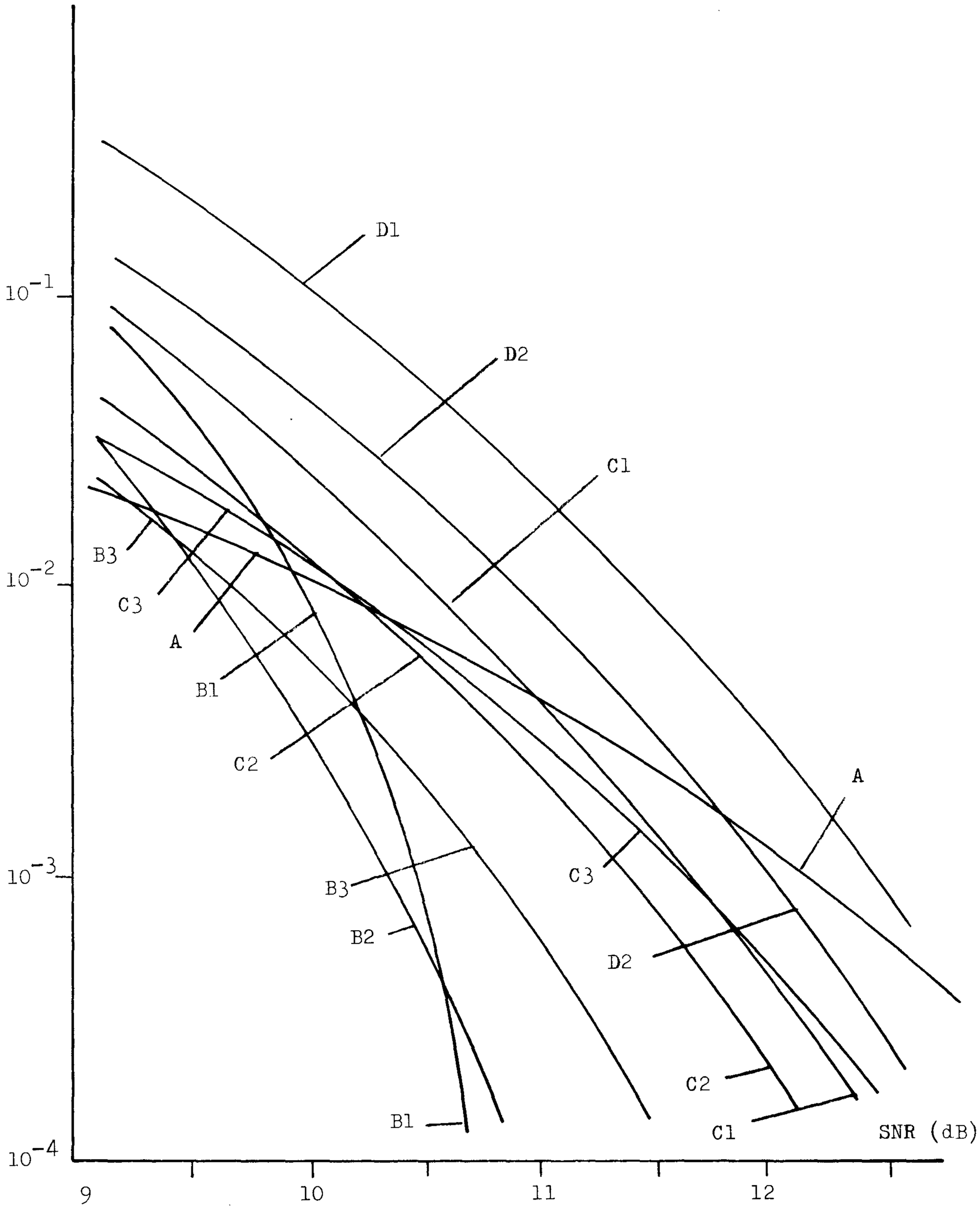


Fig. 7.4 Variation of error rate with signal to noise ratio for the various systems with coder 1 (Table 7.2) operating over channel B (Table 6.3).

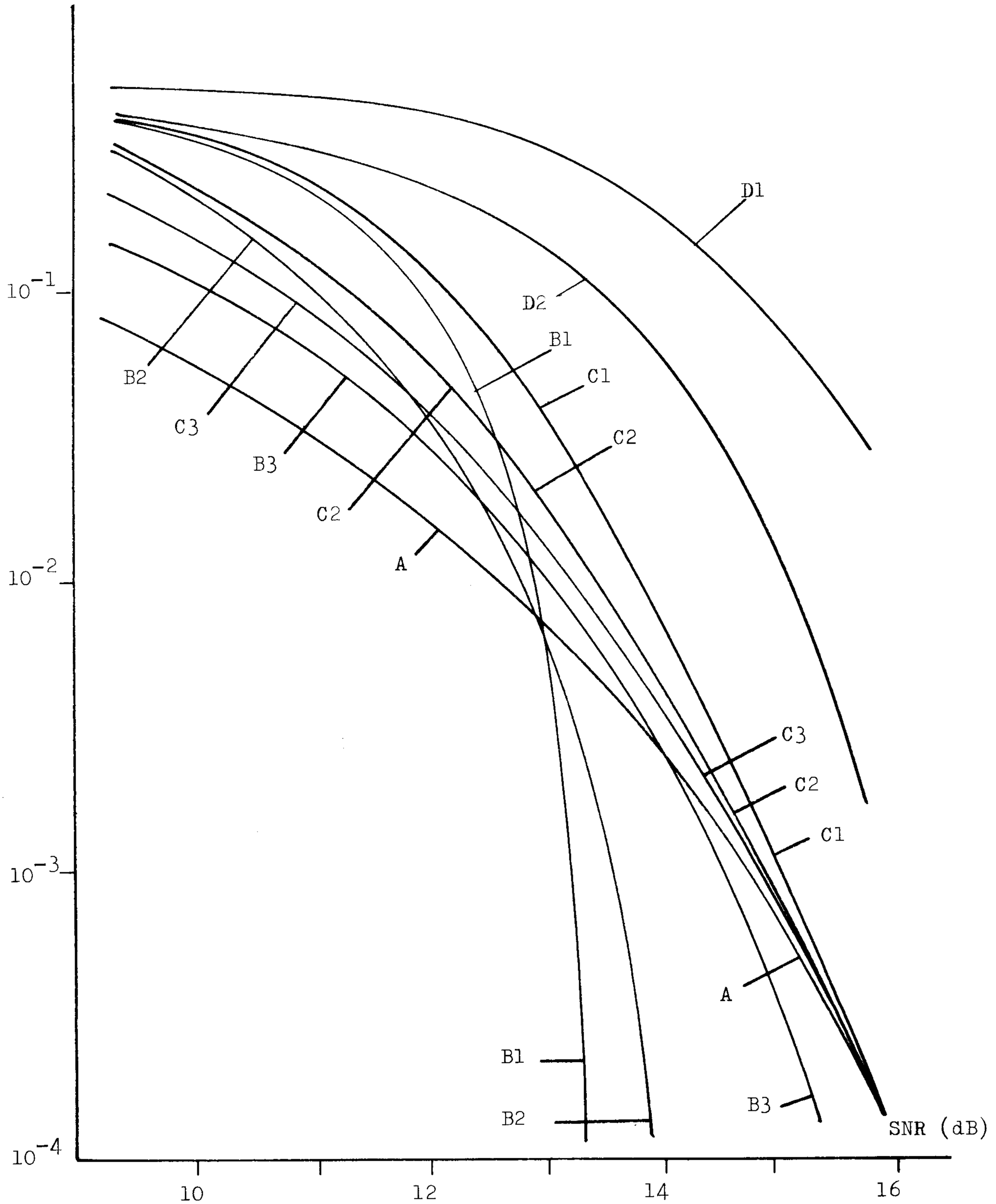


Fig. 7.5 Variation of error rate with signal to noise ratio for the various systems with coder 1 (Table 7.2) operating over channel C (Table 6.3).

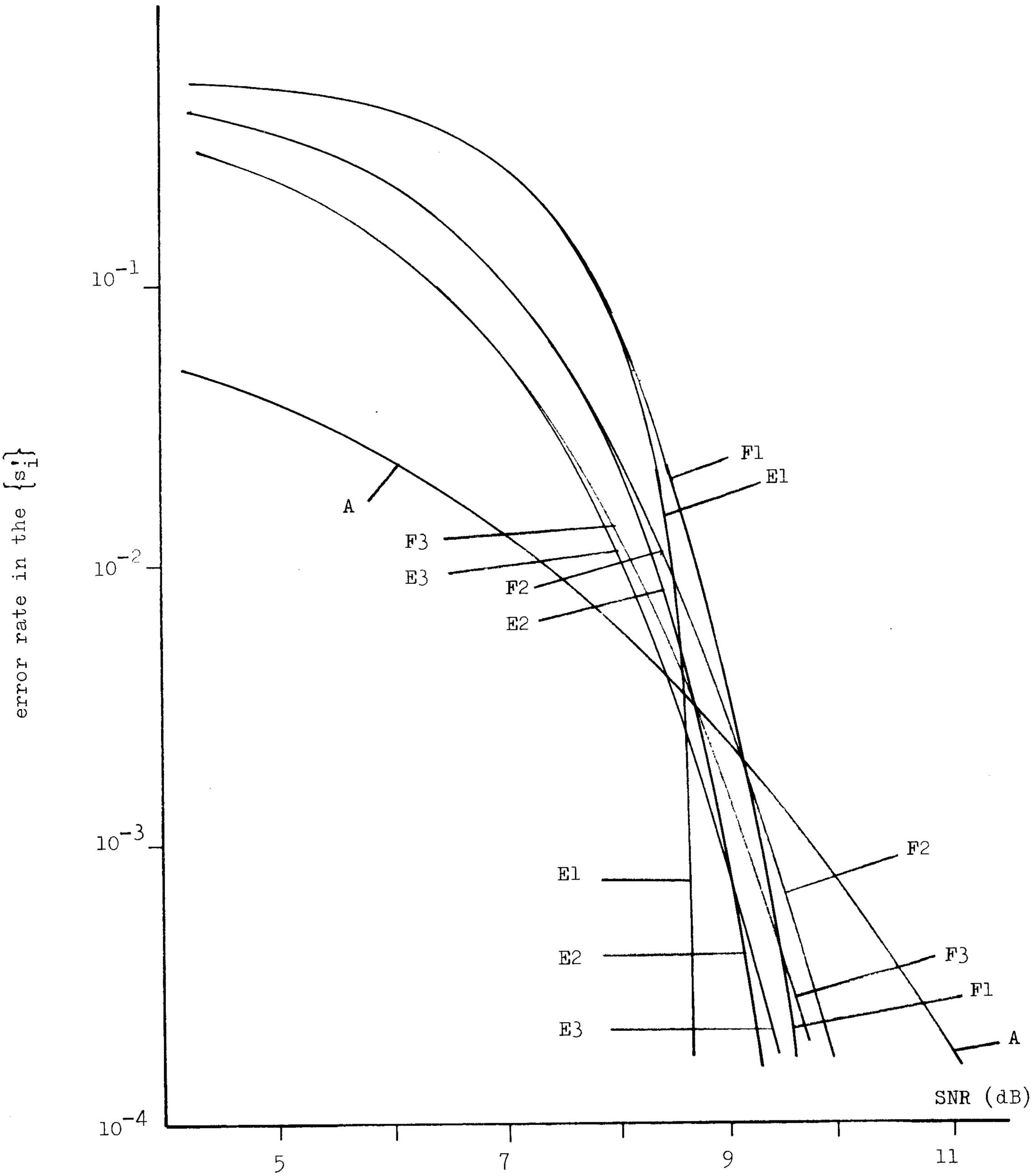


Fig. 7.6 Variation of error rate with signal to noise ratio for the various systems with coder 2 (Table 7.2) operating over channel A (Table 6.3).

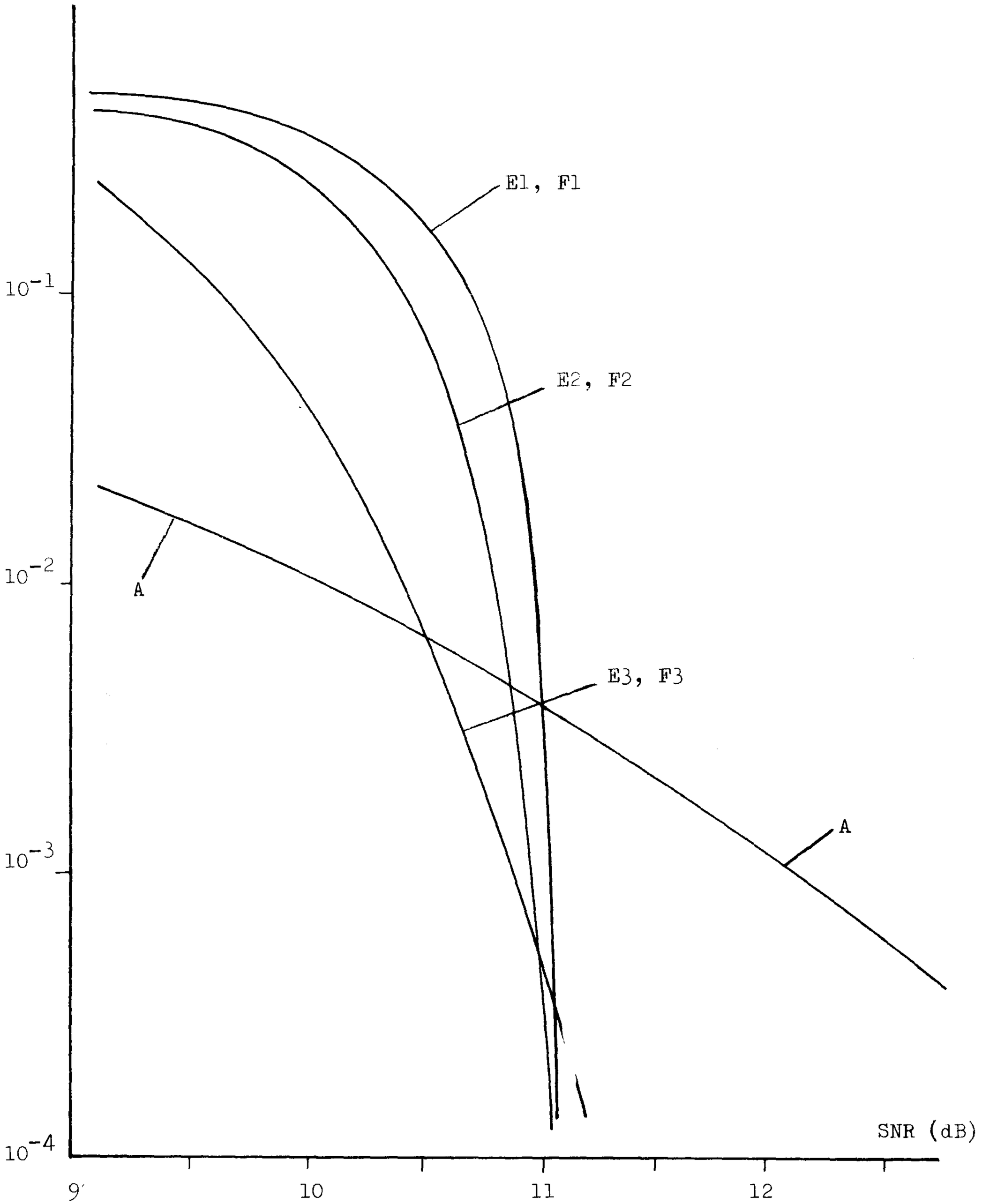


Fig. 7.7 Variation of error rate with signal to noise ratio for the various systems with coder 2 (Table 7.2) operating over channel B (Table 6.3).

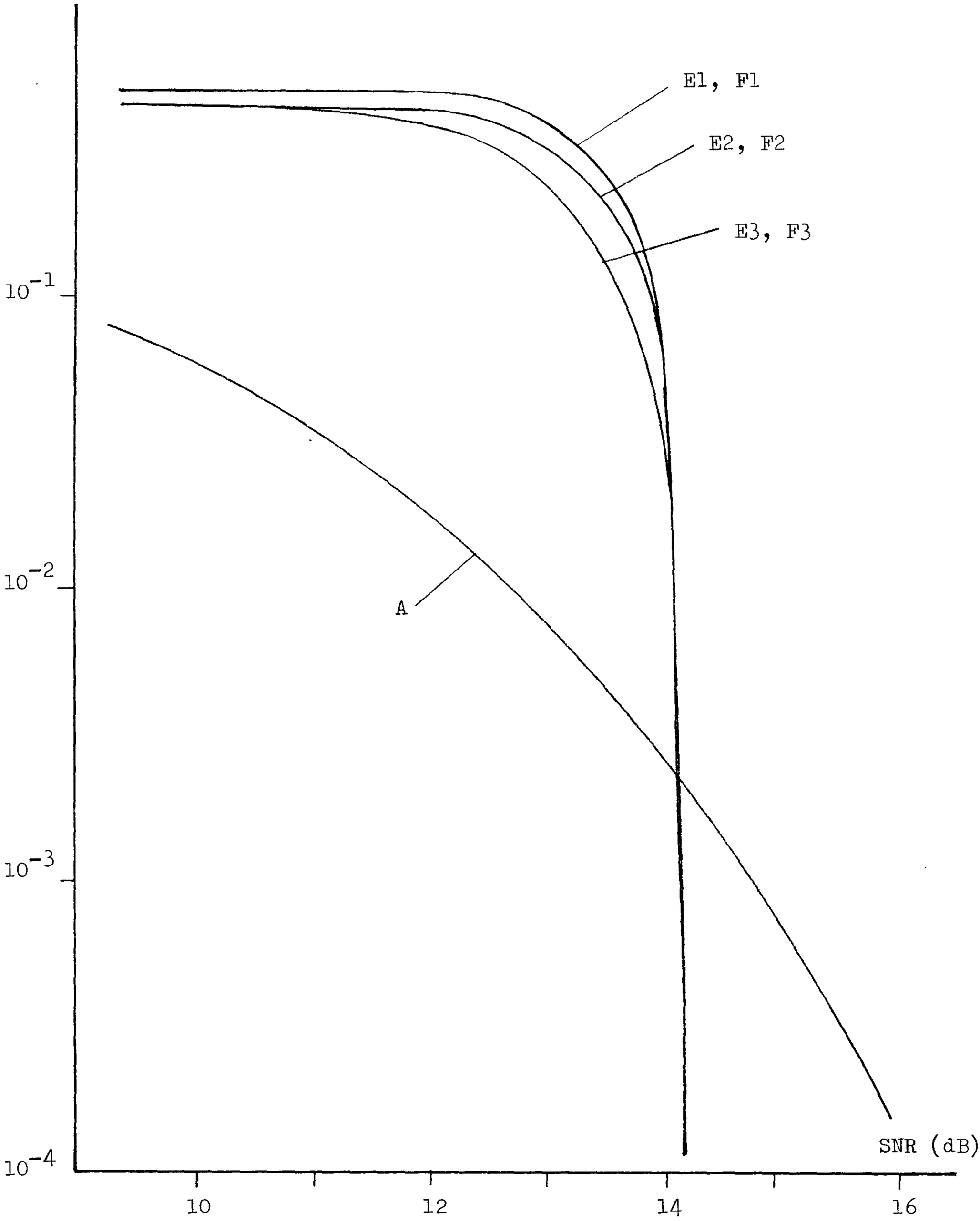


Fig. 7.8 Variation of error rate with signal to noise ratio for the various systems with coder 2 (Table 7.2) operating over channel C (Table 6.3).

7.8 Assessment of Systems

Figs. 7.3 - 7.5 show the tolerances to noise of the coded systems with coder 1 operating over the channels A, B, and C (Table 6.3). When m = 8 and n = 16 (see Table 7.2), the coded systems here appear to have relatively poor performances when operating over channel C which introduces very severe amplitude distortion. Increasing the value of m to 32 generally moves the curves of these systems to the left except for those systems operating over channel A where the improvements are observed to occur only at high error rates. When the value of n is increased from 16 to 64 while keeping m at 32, a significant improvement in the tolerance to noise at low error rates is observed for each of the coded systems here. This appears to suggest that, delay detection is an important factor that determines the performance of the coded systems studied here, bearing in mind that n sampling intervals is the delay in the detection/decoding process here.

Figs. 7.6 - 7.8 show the tolerances to noise of the coded systems with coder 2 operating over the channels A, B, and C (Table 6.3). As it appears, there is no significant difference in the tolerances to noise for n = 64 and 128 in the coded systems here. It therefore follows that, for a given value of m used in the coded systems here, there is a maximum value of n beyond which no significant improvement can be achieved in the tolerance to noise of the system. For the given values of m and n tested here, the performance of the coded system with coder 2 is generally observed to be poorer than that of the coded system with coder 1.

An important observation for the coded systems studied here is that, increasing the memory or the constraint length of the code shifts the curves to the left but steepens the slopes of them so that the

system having a larger memory may do better at low error rates while being more inferior at high error rates. This is in agreement with the basic principle of a coded system that employs an error correcting code in the system, and can be explained as follows. The effect of using an error correcting code that has a larger (or longer) memory is that, a longer stream of data digits is now being correlated. This necessarily implies that, errors tend to propagate more severely at high error rates, whereas more errors can be corrected at low error rates, when the coded system uses an error correcting code (such as the convolutional code used in the investigation here) that has a larger memory.

For the larger values of m, n, and m_e (memory) tested here, the cross-over point at which there is no difference in the tolerance to noise between the coded and the uncoded systems is generally at an error rate close to 10^{-2}, the uncoded system giving a better tolerance to noise than the coded system at higher error rates. As the error rate is reduced from 10^{-2}, the coded system gains an increasing advantage in tolerance to noise over the uncoded system, the advantage being typically around 3 dB at an error rate of 10^{-4}, for the better coded systems, and increasing rapidly as the error rate is further reduced. At an error rate of 10^{-4}, the best system appears to be B1 (Table 7.2) where coder 1 with $m_e = 7$ is used with m = 32 and n = 64 for the coded system. The performance of the coded system may be improved yet further by using larger values of m and n, but with an increase in the equipment complexity involved in the system.

Thus, the coded system with coder 1 developed in this chapter appears to be particularly promising in those applications where a

binary data signal is being transmitted at a rate close to the Nyquist rate of the given channel, and where it is required to reduce the error rate in the detected binary data values from around 10^{-4} to an extremely low value.

CHAPTER 8

CODED SYSTEM FOR 16-POINT QAM SIGNALS

8.1 Introduction

This chapter is concerned with the study of the coded system which converts the 16-point QAM (Quadrature Amplitude Modulation) signals into the appropriate 256-point or 64-point QAM signals for transmission, assuming the data-transmission system described in section 6.2. The coded system to be studied here is a development of the coded system developed in chapter 7. Since coder 1 (that involves no interleaving scheme) in chapter 7 appears to be the more promising coder without involving an excessive amount of operations and storage in the system, no interleaving scheme will be considered in this chapter. Three coding schemes each of which involves the use of a rate $\frac{1}{2}$ or $\frac{2}{3}$ convolutional code have been considered in the study here. Section 8.2 gives an account of the basic assumptions for the investigation considered in this chapter. Sections 8.3 - 8.5 give the corresponding descriptions of the three coders developed in this chapter, while section 8.6 is devoted to the descriptions of the detector used in the system here. The tolerance to Gaussian noise of the coded system is here compared with that of the uncoded system by using computer simulation tests, and the results and discussions are given in sections 8.8 and 8.9 respectively.

8.2 Basic Assumptions

The model of the data-transmission system for the coded system studied in this chapter has been described in section 6.2. The coder in Fig. 6.1 converts the binary data digits $\{s_i\}$ into the appropriate

256-point QAM or 64-point QAM coded symbols $\{q_i\}$ for transmission, while keeping the same symbol (signal element) transmission rate as the original uncoded 16-point QAM signals. The $\{s_i\}$ here are assumed to be statistically independent and equally likely to have any of the two values 0 and 1. The average transmitted energy per signal element is assumed to be equal to the mean square value of the $\{q_i\}$, where q_i is the coded symbol carried by the ith transmitted signal element here. The baseband channel (Fig. 6.1) here is assumed to be such that all zeros (or roots) of the z-transform of the channel sampled impulse response lie inside or on the unit circle in the z-plane. This assumption can always be realised by inserting, at the detector (Fig. 6.1) input, a linear filter that acts as a 'whitened matched-filter' so that the sampled impulse response of the channel and filter now has a z-transform with all the zeros lying inside or on the unit circle in the z-plane and the detector then operates on this sampled impulse response.(C42) For the purpose of study here, the arrangement of the linear filter just described is omitted and the detector here operates on the channel sampled impulse response that has all the zeros lying inside or on the unit circle. The channel sampled impulse response here has g+1 complex-valued components $y_0, y_1, \cdots, y_g$ which are assumed to be known and time-invariant. Stationary white Gaussian noise is assumed to be the only noise added at the output of the transmission path shown in Fig. 6.1, to give, at the detector input, the noise sample values $\{w_i\}$ whose real and imaginary parts are statistically independent Gaussian random variables with zero mean and a fixed variance $\frac{1}{2}N_o$. The detector (Fig. 6.1) here employs a joint near-maximum likelihood detection/decoding process, and it operates on the received sample values $\{r_i\}$ (eqn. 6.3) to give at its output the detected values of the data digits $\{s_i\}$. The quantities q_i, w_i, and r_i are, in general, complex-valued quantities here.

8.3 Coder 1

This coder encodes the binary data digits $\{s_i\}$ into the appropriate 256-point QAM coded symbols $\{q_i\}$ for transmission. The schematic diagram of the coder here is shown in Fig. 8.1, and its operation is described as follows. A rate $\frac{1}{2}$ binary convolutional coder is first used to code the $\{s_i\}$ into binary coded digits $\{v_i^{(e)}\}$ where

$$v_i^{(e)} = \sum_{h=0}^{\nu-1}{}' \, s_{i-h} g_h^{(e)} \qquad (8.1)$$

for $e = 1, 2$. In eqn. 8.1, the $\{g_h^{(e)}\}$ are the components of the code generators $\{G^{(e)}\}$ (eqn. 6.12) of the convolutional code with constraint length ν, and $\sum'$ is a modulo-2 adder. Eqn. 8.1 is in fact identical to eqn. 7.1. Since modulo-2 addition is used here, the value of $v_i^{(e)}$ is either 0 or 1. Every 8 neighbouring values of $\{v_i^{(e)}\}$ are here combined to give the 8-component vector

$$V_i = \left[v_{4i-3}^{(1)} \; v_{4i-3}^{(2)} \; v_{4i-2}^{(1)} \; v_{4i-2}^{(2)} \; v_{4i-1}^{(1)} \; v_{4i-1}^{(2)} \; v_{4i}^{(1)} \; v_{4i}^{(2)} \right] \qquad (8.2)$$

A Gray coder (Fig. 8.1) is then used to map the vector V_i into a suitable 256-point QAM coded symbol q_i for transmission. Being a 256-point QAM signal, q_i may be expressed as the complex-valued quantity

$$q_i = a_i + jb_i \qquad (8.3)$$

and $a_i, b_i = \pm 1/\sqrt{170}, \pm 3/\sqrt{170}, \pm 5/\sqrt{170}, \pm 7/\sqrt{170},$
$\pm 9/\sqrt{170}, \pm 11/\sqrt{170}, \pm 13/\sqrt{170}, \pm 15/\sqrt{170}$ (8.4)

where $j = \sqrt{-1}$, and a_i and b_i are the in-phase component and the quadrature component of q_i respectively. The factor $\sqrt{170}$ is included

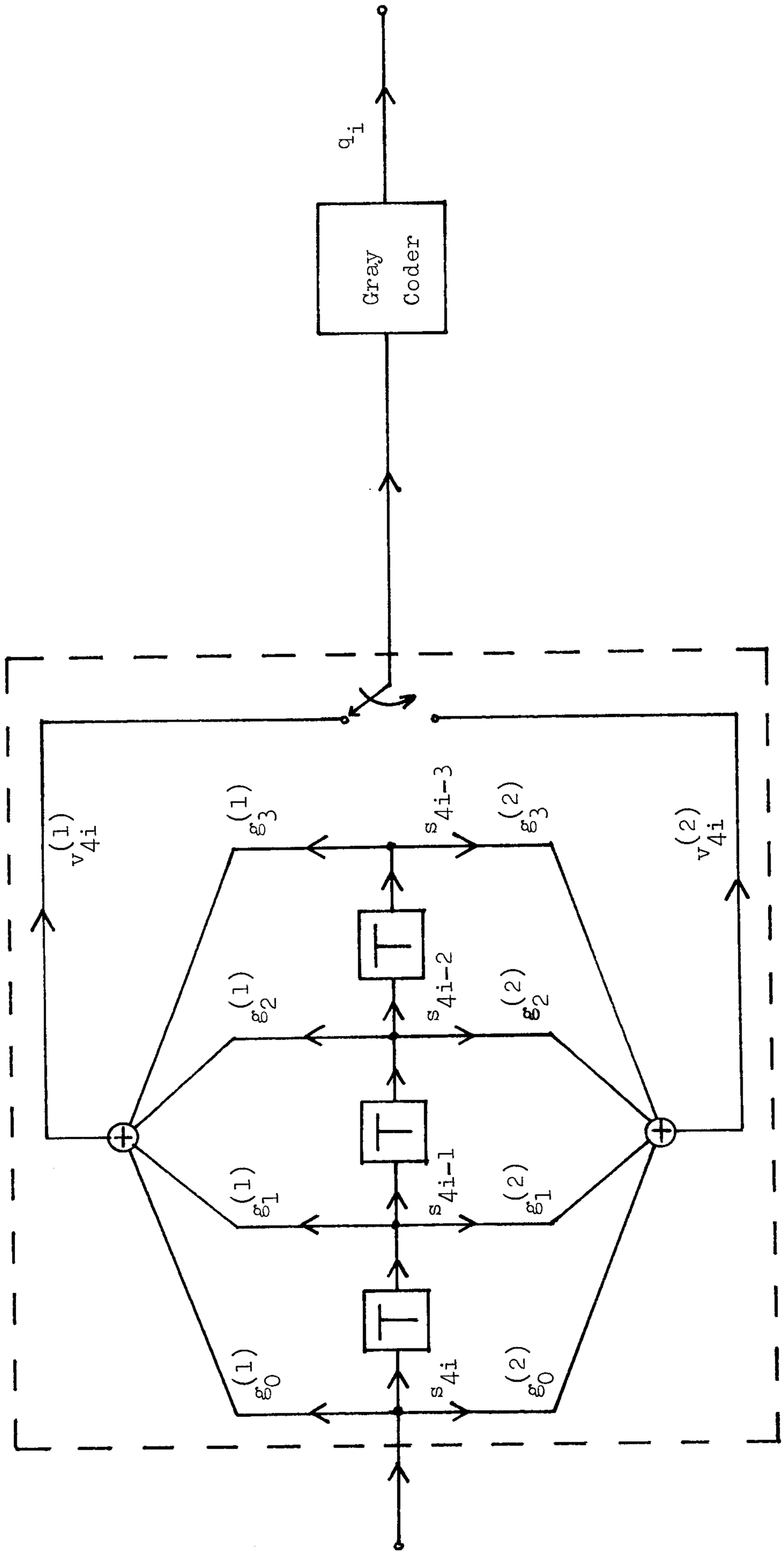


Fig. 8.1 An example of the schematic diagram of coder 1 with $\nu = 4$. ν is the constraint length of the convolutional code having the code generators $\{g_h^{(e)}\}$. $\oplus$ is a modulo-2 adder.

in eqn. 8.4 to ensure that the average transmitted energy per signal element, which is the mean-square value of the $\{q_i\}$, is unity. The Gray code used to convert the vector V_i (eqn. 8.2) into the coded symbol q_i here is such that, the first two components of V_i are used to determine the quadrant in which q_i should lie as is shown in Fig. 8.2. The next three components of V_i are then used to determine the absolute value of the in-phase component a_i of q_i, while the last three components of V_i are used to determine the absolute value of the quadrature component b_i of q_i, according to Table 8.1. The mapping of V_i into q_i described above is summarised in Fig. 8.3 where each possible vector of V_i is expressed as a decimal value corresponding to the binary value formed by the 8 components of V_i (eqn. 8.2). It can be seen from Fig. 8.3 that, for any adjacent values of $\{q_i\}$ (which are separated by a distance of $2/\sqrt{170}$), the corresponding vectors $\{V_i\}$ differ in only one component.

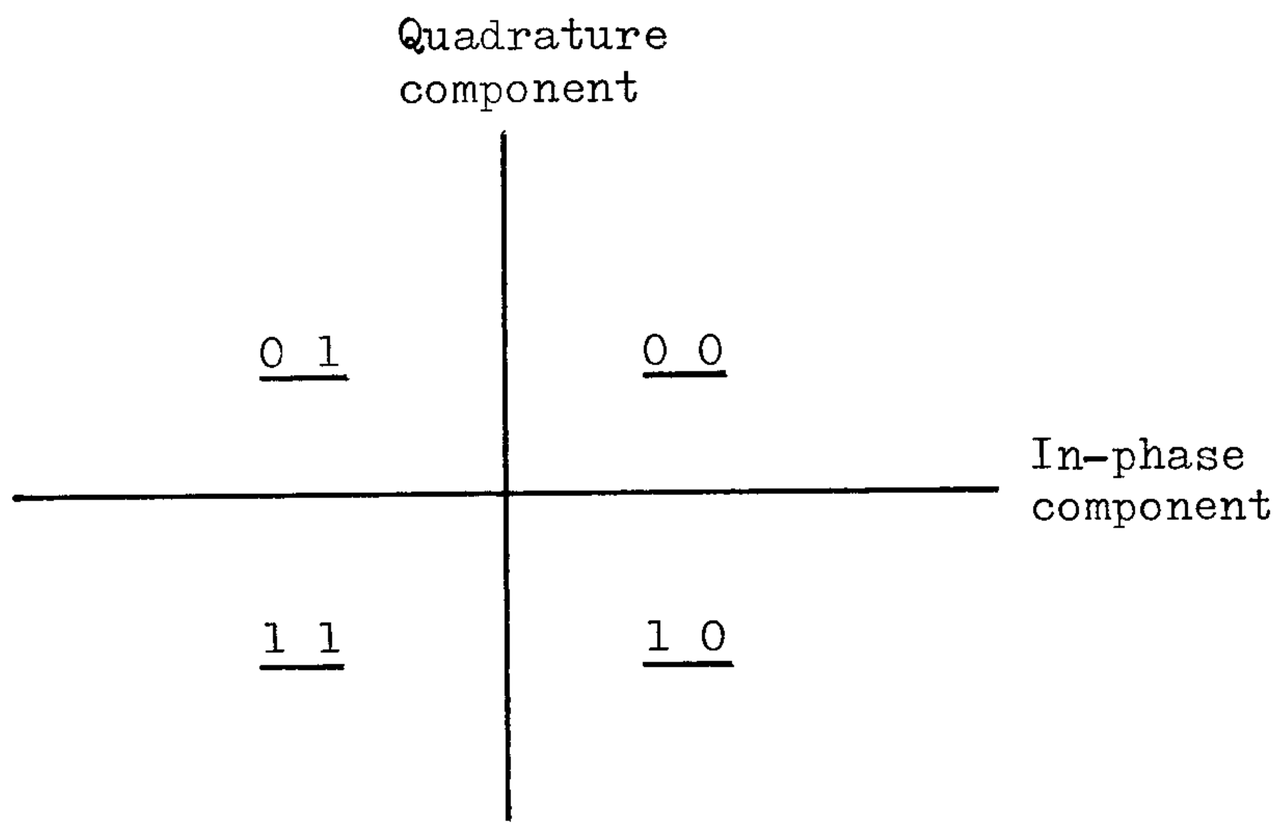


Fig. 8.2 Relationship between the first two components $v_{4i-3}^{(1)}$, $v_{4i-3}^{(2)}$ of V_i (eqn. 8.2) and the quadrant in which q_i lies.

$v^{(1)}_{4i-2}$ $v^{(2)}_{4i-2}$ $v^{(1)}_{4i-1}$ (or $v^{(2)}_{4i-1}$ $v^{(1)}_{4i}$ $v^{(2)}_{4i}$)	$\lvert a_i \rvert$ (or $\lvert b_i \rvert$)
0 0 0	$1/\sqrt{170}$
0 0 1	$3/\sqrt{170}$
0 1 1	$5/\sqrt{170}$
0 1 0	$7/\sqrt{170}$
1 1 0	$9/\sqrt{170}$
1 1 1	$11/\sqrt{170}$
1 0 1	$13/\sqrt{170}$
1 0 0	$15/\sqrt{170}$

Table 8.1 Relationship between the last six components of V_i (eqn. 8.2) and the absolute values of the in-phase component a_i and quadrature component b_i of the coded symbol q_i for coder 1.

100	108	124	116	84	92	76	68	4	12	28	20	52	60	44	36
101	109	125	117	85	93	77	69	5	13	29	21	53	61	45	37
103	111	127	119	87	95	79	71	7	15	31	23	55	63	47	39
102	110	126	118	86	94	78	70	6	14	30	22	54	62	46	38
98	106	122	114	82	90	74	66	2	10	26	18	50	58	42	34
99	107	123	115	83	91	75	67	3	11	27	19	51	59	43	35
97	105	121	113	81	89	73	65	1	9	25	17	49	57	41	33
96	104	120	112	80	88	72	64	0	8	24	16	48	56	40	32
224	232	248	240	208	216	200	192	128	136	152	144	176	184	168	160
225	233	249	241	209	217	201	193	129	137	153	145	177	185	169	161
227	235	251	243	211	219	203	195	131	139	155	147	179	187	171	163
226	234	250	242	210	218	202	194	130	138	154	146	178	186	170	162
230	238	254	246	214	222	206	198	134	142	158	150	182	190	174	166
231	239	255	247	215	223	207	199	135	143	159	151	183	191	175	167
229	237	253	245	213	221	205	197	133	141	157	149	181	189	173	165
228	236	252	244	212	220	204	196	132	140	156	148	180	188	172	164

Fig. 8.3 Mapping of the 8-component vector V_i (eqn. 8.2) into the 256-point QAM signal q_i for coder 1. The 8 binary digits in V_i are represented by their corresponding decimal value here.

8.4 Coder 2

Coder 2 is a modification of Coder 1 in that, it uses the same rate $\frac{1}{2}$ binary convolutional coder but a different Gray coder to convert the binary data digits $\{s_i\}$ into the appropriate 64-point (instead of 256-point) QAM coded symbols $\{q_i\}$ for transmission. This coder is otherwise the same as coder 1. Thus, coder 2 has the same schematic diagram as that shown in Fig. 8.1 and it operates as follows. The rate $\frac{1}{2}$ convolutional coder (Fig. 8.1) is first used to code the $\{s_i\}$ into the coded digits $\{v_i^{(e)}\}$ (eqn. 8.1) in the same way as is done in coder 1. A 6-component vector V_i is then formed from every 8 neighbouring coded digits $\{v_i^{(e)}\}$, where

$$V_i = \left[v_{4i-3}^{(1)} \; v_{4i-3}^{(2)} \; v_{4i-2}^{(1)} \; v_{4i-2}^{(2)} \; v_{4i-1}^{(2)} \; v_{4i}^{(1)} \right] \qquad (8.5)$$

A Gray code is then used to convert the vector V_i here into a 64-point QAM coded symbol q_i, where q_i has a complex value given by $a_i + jb_i$ and

$$a_i, \; b_i = \pm 1/\sqrt{42}, \; \pm 3/\sqrt{42}, \; \pm 5/\sqrt{42}, \; \pm 7/\sqrt{42} \qquad (8.6)$$

The factor $\sqrt{42}$ is included here so that the average transmitted energy per signal element is equal to unity, bearing in mind that the average transmitted energy per signal element is here assumed to be equal to the mean-square value of the $\{q_i\}$. The conversion of the vector V_i given by eqn. 8.5 into the 64-point QAM symbol q_i is here carried out as follows. The first two components $v_{4i-3}^{(1)}$ and $v_{4i-3}^{(2)}$ of V_i here are first used to determine the quadrant in which q_i should lie. This is carried out according to the arrangement shown in Fig. 8.2. The

next two components of V_i are then used to determine the absolute value of the in-phase component a_i of q_i, while the last two components of V_i are used to determine the absolute value of the quadrature component b_i of q_i, according to Table 8.2. The mapping of the vector V_i (eqn. 8.5) into the corresponding q_i described above is summarised in Fig. 8.4. It can be seen from Figs. 8.3, 8.4 and Tables 8.1, 8.2 that, coder 2 effectively operates by reducing every four neighbouring signal points in Fig. 8.3 (which contains the 256 possible coded symbols for coder 1) into just one signal point in Fig. 8.4 (which contains the 64 possible coded symbols for coder 2), as is illustrated in Fig. 8.5.

Thus, the system with coder 2 has the advantage of having fewer possible values for the transmitted symbols $\{q_i\}$ and hence a probably higher tolerance to noise, as compared to the system with coder 1. However, the penalty that has to be paid here is that, there are now less redundancies in the coded symbols $\{q_i\}$ so that the effective minimum free distance of the convolutional code may now become smaller leading to a reduction in the error correction capability of the coded system here.

$v_{4i-2}^{(1)}$ $v_{4i-2}^{(2)}$ (or $v_{4i-1}^{(2)}$ $v_{4i}^{(1)}$)	$\lvert a_i \rvert$ (or $\lvert b_i \rvert$)
0 0	$1/\sqrt{42}$
0 1	$3/\sqrt{42}$
1 1	$5/\sqrt{42}$
1 0	$7/\sqrt{42}$

Table 8.2 Relationship between the last four components of V_i (eqn. 8.5) and the absolute values of the in-phase component a_i and quadrature component b_i of the coded symbol q_i for coder 2.

011010	011110	010110	010010	000010	000110	001110	001010
26	30	22	18	2	6	14	10
011011	011111	010111	010011	000011	000111	001111	001011
27	31	23	19	3	7	15	11
011001	011101	010101	010001	000001	000101	001101	001001
25	29	21	17	1	5	13	9
011000	011100	010100	010000	000000	000100	001100	001000
24	28	20	16	0	4	12	8
111000	111100	110100	110000	100000	100100	101100	101000
56	60	52	48	32	36	44	40
111001	111101	110101	110001	100001	100101	101101	101001
57	61	53	49	33	37	45	41
111011	111111	110111	110011	100011	100111	101111	101011
59	63	55	51	35	39	47	43
111010	111110	110110	110010	100010	100110	101110	101010
58	62	54	50	34	38	46	42

Fig. 8.4 Mapping of the 6-component vector V_i (eqn. 8.5) into the 64-point QAM signal q_i for Coder 2.

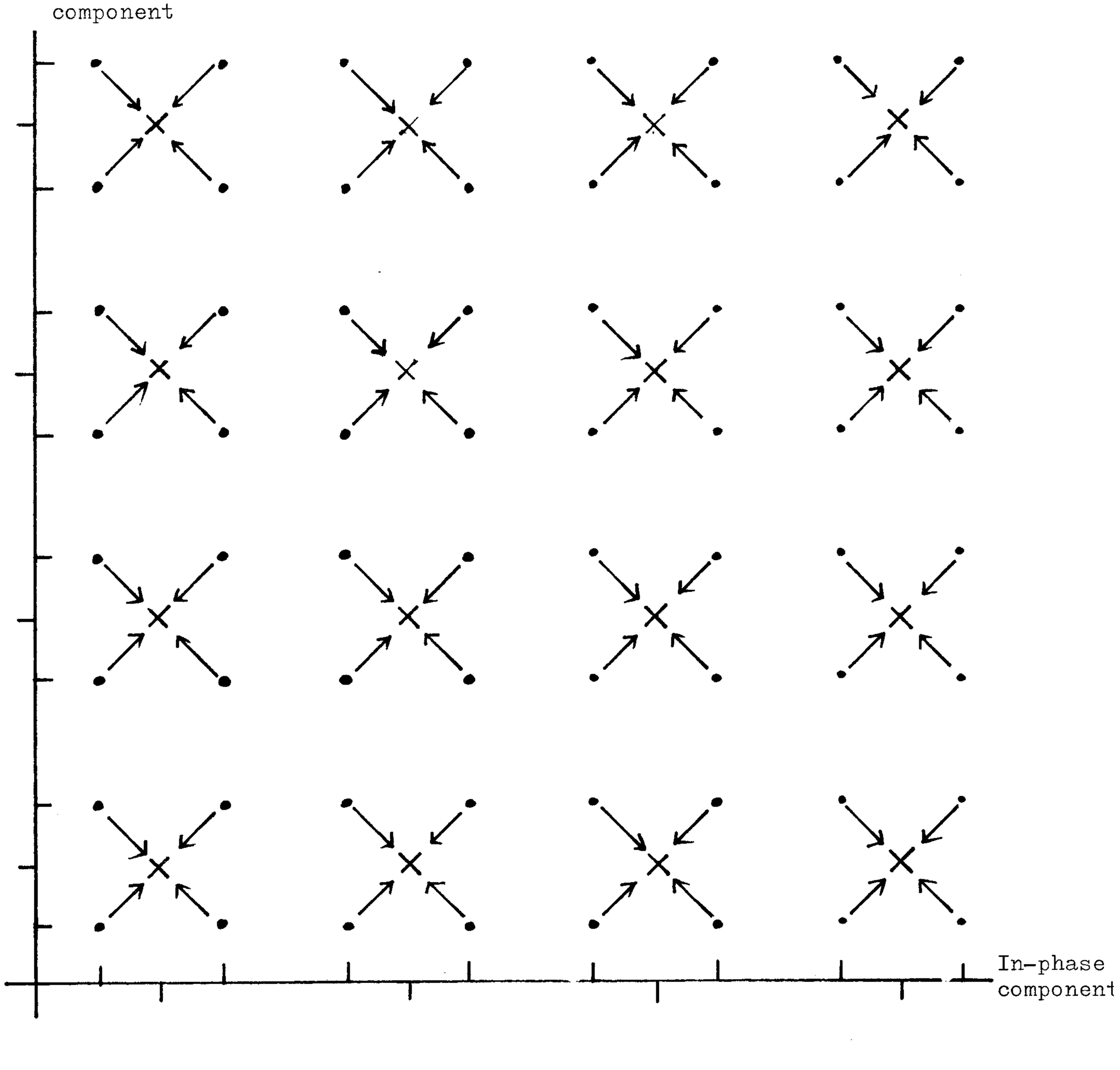


Fig. 8.5 Relationship between the coded symbols for coder 1 (Fig. 8.3) and the coded symbols for coder 2 (Fig. 8.4) in the first quadrant of the corresponding QAM signal constellations of these coded symbols. ● is for coder 1, and ✕ is for coder 2. Every four neighbouring ● correspond to one ✕ here.

8.5 Coder 3

Coder 3 employs a coding scheme which involves the use of a rate $\frac{2}{3}$ binary convolutional code and a suitable Gray code in converting the original uncoded 16-point QAM signals into the appropriate 64-point QAM signals for transmission, assuming the data-transmission system of Fig. 6.1. The schematic diagram of this coder is shown in Fig. 8.6.

In Fig. 8.6, the binary data sequence $\{s_i\}$ is coded into three streams of binary sequences $\{v_i^{(1)}\}$, $\{v_i^{(2)}\}$, and $\{v_i^{(3)}\}$ through the rate $\frac{2}{3}$ binary convolutional coder which consists of three code generators $G^{(1)}$, $G^{(2)}$, and $G^{(3)}$, where the $\{v_i^{(e)}\}$ and the $\{G^{(e)}\}$ are as defined by eqns. 8.1 and 6.12 respectively. The data digits $\{s_i\}$ in Fig. 8.6 are shifted two places at a time to the right so that after shifting twice, four new digits of $\{s_i\}$ have entered the coder and six corresponding coded digits $\{v_i^{(e)}\}$ are produced at the output of the convolutional coder, to give the 6-component vector V_i where

$$V_i = \left[\begin{array}{cccccc} v_{4i-2}^{(1)} & v_{4i-2}^{(2)} & v_{4i-2}^{(3)} & v_{4i}^{(1)} & v_{4i}^{(2)} & v_{4i}^{(3)} \end{array} \right] \qquad (8.7)$$

The Gray coder (Fig. 8.6) then maps the vector V_i into a 64-point QAM coded symbol q_i in the same way as is done in coder 2. That is, the first two components of V_i are used to determine the quandrant in which q_i should lie according to the arrangement shown in Fig. 8.2. The next two components of V_i are then used to determine the absolute value of the in-phase component a_i of q_i, while the last two components of V_i are used to determine the absolute value of the quadrature component b_i of q_i, according to Table 8.2. The resultant mapping of V_i to q_i is summarised in Fig. 8.4.

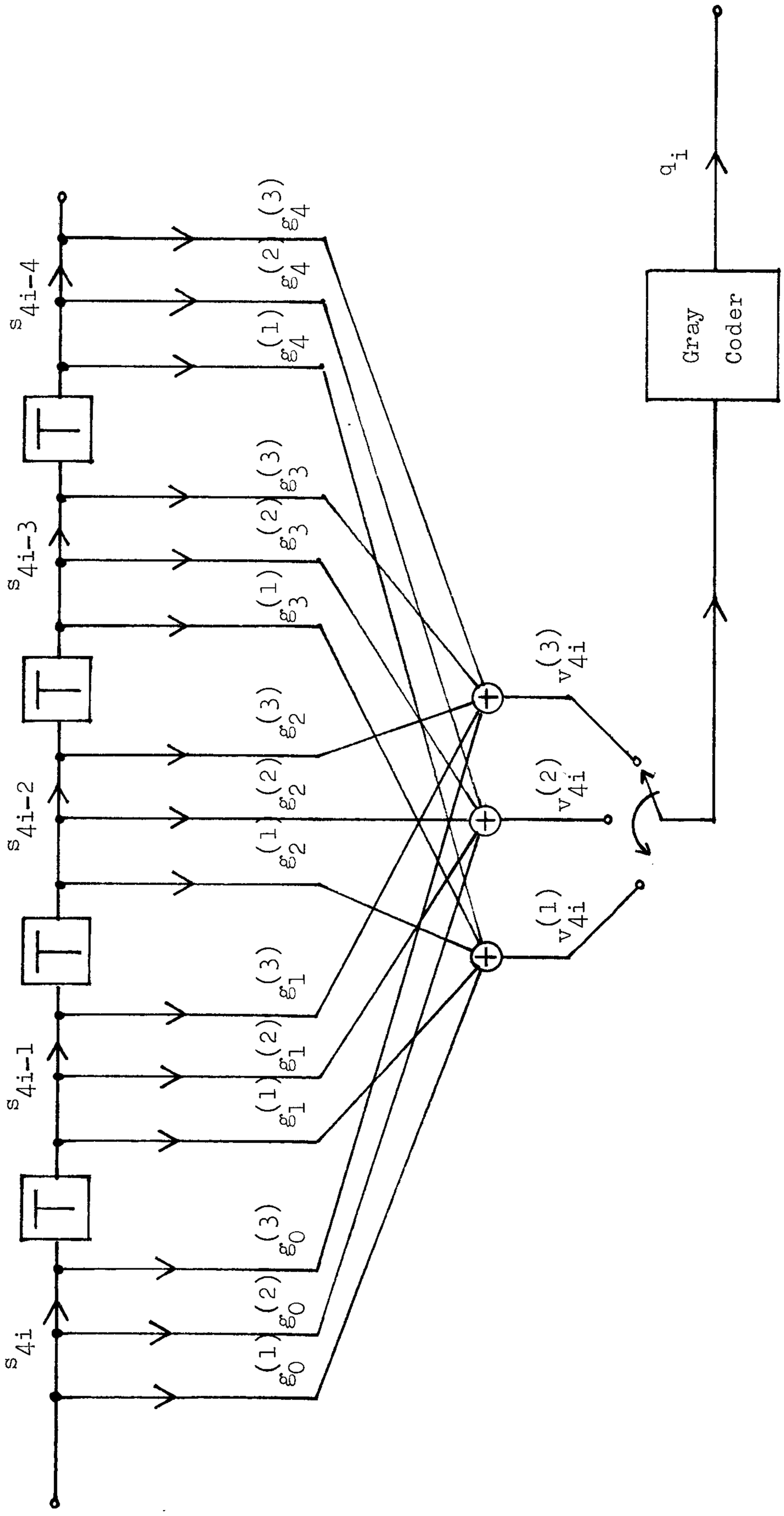


Fig. 8.6 An example of the schematic diagram of coder 3 with $\nu = 5$. ν is the constraint length of the rate-$\frac{2}{3}$ convolutional code having the code generators $\{g_h^{(e)}\}$. ⊕ is a modulo-2 adder.

One possible advantage of coder 3 over coder 2 is that, it uses an apparently more appropriate convolutional code in converting every four data digits $\{s_i\}$ (corresponding to a 16-point QAM uncoded signal) into one 64-point QAM coded symbol q_i, bearing in mind that a rate $\frac{1}{2}$ convolutional code is used in coder 2 which is apparently more appropriate in converting a 16-point QAM signal into a 256-point QAM signal. The drawback of the coding scheme used in coder 3 is that, the optimum free distance (see section 6.5) of the rate $\frac{2}{3}$ convolutional code (which is being used in coder 3) increases more slowly with the constraint length of the code as compared with the rate $\frac{1}{2}$ convolutional code, so that more components are required to be used in the code generators here to achieve the same free distance. Furthermore, the implementation of a rate $\frac{2}{3}$ convolutional coder also appears to be slightly more complex than that of a rate $\frac{1}{2}$ convolutional coder as can be seen from Figs. 8.1 and 8.6.

8.6 Detector

The detector (Fig. 6.1) here employs a joint near-maximum likelihood detection/decoding process, and it operates on the received sample values $\{r_i\}$ (eqn. 6.3) to give at its output the detected values for the data digits $\{s_i\}$. The received sample value r_i at time $t = iT$ is, from eqn. 6.3, given by

$$r_i = \sum_{h=0}^{g} q_{i-h} y_h + w_i \qquad (8.8)$$

where $\{q_i\}$ are the transmitted QAM coded symbols, $\{w_i\}$ are the noise components, and $y_0, y_1, \cdots, y_g$ are the g+1 components of the channel sampled impulse response. The model of the detector here is shown in Fig. 8.7, and its operation is described as below.

The near-maximum likelihood detection/decoding processor in Fig. 8.7 operates on its input samples $\{r'_i\}$ to give the finally detected data digits $\{s'_i\}$ and the corresponding detected coded symbols $\{q'_i\}$, where $s'_{4i-3}, s'_{4i-2}, s'_{4i-1}, s'_{4i}$ and their corresponding detected coded symbol q'_i are here determined after the receipt of r'_{i+n} and $n < g$. The operation of this processor will shortly be described. The intersymbol interference canceller in Fig. 8.7 operates by removing from the received samples $\{r_i\}$ (eqn. 8.8), detected values of all components involving coded symbols $\{q_i\}$ whose final detected values $\{q'_i\}$ have already been determined. That is, the intersymbol interference canceller operates on r_i to give the complex-valued quantity

$$r'_i = r_i - \sum_{h=n+1}^{g} q'_{i-h} y_h \qquad (8.9)$$

Assuming for the moment that $q'_{i-h} = q_{i-h}$ for $h = n+1, n+2, \cdots, g$

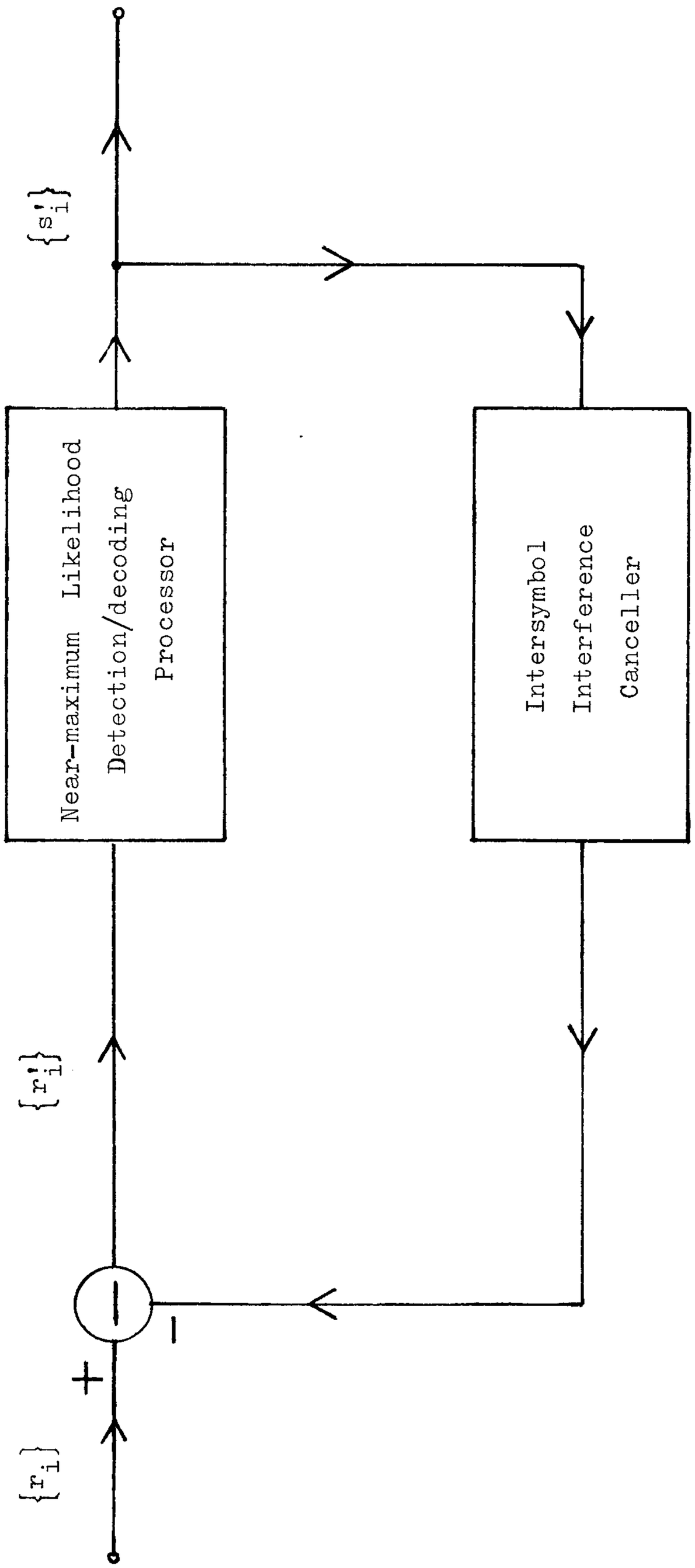


Fig. 8.7 Model of the detector (Fig. 6.1).

eqn. 8.9 now reduces to

$$r'_i = \sum_{h=0}^{n} q_{i-h} y_h + w_i \qquad (8.10)$$

where the $\{y_j\}$ are the components of the channel sampled impulse response, and the $\{w_j\}$ are the noise components here. It can be seen from eqns. 8.8 and 8.10 that, the effective number of components of the sampled impulse response of the channel has now been reduced, from g+1, to n+1, so that the detection/decoding process is now greatly simplified when $n \ll g$.

The operation of the near-maximum likelihood detection/decoding processor shown in Fig. 8.7 is now described as follows. Just prior to the receipt of the sample r'_i at the processor input, the processor holds in store m (4n)-component vectors $\{X_{i-1}\}$ where

$$X_{i-1} = \left[x_{4(i-n)-3} \quad x_{4(i-n)-2} \quad \cdots \quad x_{4(i-1)} \right] \qquad (8.11)$$

and x_h may have any of the two possible values (0 and 1) of the data digit s_h for any possible value of h. Associated with each stored vector X_{i-1}, are stored the corresponding value of cost C_{i-1} which will shortly be considered, and the corresponding n quantities q'_{i-n}, q'_{i-n+1}, ·· , q'_{i-1}, where q'_h may have any of the possible values of the coded symbol q_h. The evaluations of the values of the $\{q'_h\}$ here will shortly be considered.

On the receipt of r'_i, each vector X_{i-1} is expanded into 16 (4n+4)-component vectors $\{P_i\}$, where

$$P_i = \left[x_{4(i-n)-3} \quad x_{4(i-n)-2} \quad \cdots \quad x_{4i} \right] \qquad (8.12)$$

The first 4n components of P_i are as in the original vector X_{i-1}, and

the last 4 components $x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}$ have the 16 possible values of the data digits $s_{4i-3}, s_{4i-2}, s_{4i-1}, s_{4i}$, bearing in mind that s_h may have any of the two possible values 0 and 1 for $h > 0$. Each vector P_i here is also associated with the same set of quantities $q'_{i-n}, q'_{i-n+1}, \cdots, q'_{i-1}$ as the original vector X_{i-1}. The value of q'_i for each of the 16m expanded vectors $\{P_i\}$ is now evaluated as follows. When coder 1 is used in the system, the processor here (Fig. 8.6) first forms the 8-component vector

$$U_i = \left[u^{(1)}_{4i-3} \; u^{(2)}_{4i-3} \; u^{(1)}_{4i-2} \; u^{(2)}_{4i-2} \; u^{(1)}_{4i-1} \; u^{(2)}_{4i-1} \; u^{(1)}_{4i} \; u^{(2)}_{4i} \right] \quad (8.13)$$

where the quantity $u^{(e)}_j$ is evaluated as

$$u^{(e)}_j = \sum_{h=0}^{\nu-1}{}' \, x_{j-h} g^{(e)}_h \quad (8.14)$$

for $e = 1, 2,$ and the quantities ν and $\{g^{(e)}_h\}$ are as defined in eqn. 8.1. The symbol $\sum'$ in eqn. 8.14 is a modulo-2 adder. The vector U_i (eqn. 8.13) is then used to determine the value of q'_i for the given vector P_i, where the relationship between q'_i and U_i is identical to that between q_i and V_i in Fig. 8.3. When coder 2 is used in the system, the processor (Fig. 8.6) forms the 6-component vector

$$U_i = \left[u^{(1)}_{4i-3} \; u^{(2)}_{4i-3} \; u^{(1)}_{4i-2} \; u^{(2)}_{4i-2} \; u^{(2)}_{4i-1} \; u^{(1)}_{4i} \right] \quad (8.15)$$

where the $\{u^{(e)}_j\}$ here are as defined in eqn. 8.14. The value of q'_i for the given vector P_i is then determined from this vector (eqn. 8.15), where the relationship between q'_i and U_i here is identical to that between q_i and V_i in Fig. 8.4. When coder 3 is used in the system,

the processor (Fig. 8.6) forms the 6-component vector

$$U_i = \left[u_{4i-2}^{(1)} \; u_{4i-2}^{(2)} \; u_{4i-2}^{(3)} \; u_{4i}^{(1)} \; u_{4i}^{(2)} \; u_{4i}^{(3)} \right] \qquad (8.16)$$

where the $\{u_j^{(e)}\}$ here are as defined in eqn. 8.14. This vector is then used to determine the value of q_i' for the given vector P_i (eqn. 8.12), where the relationship between q_i' and U_i here is identical to that between q_i and V_i in Fig. 8.4. Having evaluated the value of q_i' for each of the 16m expanded vectors $\{P_i\}$, the processor (Fig. 8.6) then determines, for each of these $\{P_i\}$, the scalar quantity

$$z_i' = \sum_{h=0}^{n} q_{i-h}' y_h \qquad (8.17)$$

This is an estimate of the quantity

$$z_i = \sum_{h=0}^{n} q_{i-h} y_h \qquad (8.18)$$

which is the signal component in the sample value r_i' (eqn. 8.10). Having evaluated the value of z_i' for each of the 16m vectors $\{P_i\}$, the processor then evaluates the cost C_i for each of these $\{P_i\}$ as

$$C_i = C_{i-1} + |r_i - z_i'|^2 \qquad (8.19)$$

where $|r_i - z_i'|$ is the absolute value of the complex-valued quantity $r_i - z_i'$, and the value of C_{i-1} here has already been determined for each vector X_{i-1} in the previous detection process. It can be deduced from eqn. 8.19 that

$$C_i = \sum_{h=1}^{i} |r_h - z_h'|^2 \qquad (8.20)$$

so that the cost C_i is in fact the square of the unitary distance between the sequence of sample values, R'_i, and the estimate Z'_i of this sequence in the absence of noise, where

$$Z'_i = \begin{bmatrix} z'_1 & z'_2 & \cdots & z'_i \end{bmatrix} \qquad (8.21)$$

$$R'_i = \begin{bmatrix} r'_1 & r'_2 & \cdots & r'_i \end{bmatrix} \qquad (8.22)$$

Having evaluated the cost C_i for each of the 16m expanded vectors $\{P_i\}$, the processor then proceeds to select the vector P_i associated with the smallest cost, and it takes the values of $x_{4(i-n)-3}$, $x_{4(i-n)-2}$, $x_{4(i-n)-1}$, and $x_{4(i-n)}$ to be the corresponding values of the detected data digits $s'_{4(i-n)-3}$, $s'_{4(i-n)-2}$, $s'_{4(i-n)-1}$, and $s'_{4(i-n)}$. All vectors $\{P_i\}$ for which at least one of the first four components in the vector are not equal to the corresponding values of the four detected data digits just determined are now discarded. The first four components of each of the remaining vectors $\{P_i\}$ are then omitted to give the corresponding (4n)-component vectors $\{X_i\}$. The cost of a vector X_i here is taken to be the cost of the corresponding vector P_i. Thus, the vector X_i associated with the smallest cost is here the first selected vector. The processor then selects the second vector X_i associated with the smallest cost subject to the constraint that, only the 16 vectors $\{X_i\}$ originating from the previous first (best) vector X_{i-1} are available for the selection here. The constraint here is to ensure that the previous best vector is retained so that, in the case of severe impulsive noise where the noise samples are very large, not all the vectors originating from the previous best vector are discarded from the store. Having selected the first two vectors, the processor then proceeds to select, from the remaining vectors $\{X_i\}$, the m-2 vectors

associated with the smallest costs, to give a total of m selected vectors $\{X_i\}$. These m selected vectors are then stored together with their associated costs $\{C_i\}$. The detector is now ready for the next detection/decoding process and the process continues in this way.

To start the detection/decoding process here, the detector stores m vectors $\{X_n\}$ which are all the same and correct, a known synchronizing signal being transmitted here, and to one of these vectors it ascribes a cost value of zero, whereas to each of the remainders it ascribes a very high cost value.

In the case where no vectors of $\{X_i\}$ are available for selection during the detection/decoding process described above, a vector with arbitrary components $\{x_h\}$ is selected and its associated cost is set to a very high value so that this vector is discarded in the next detection/decoding process.

8.7 Uncoded System

The uncoded system here is a system which does not employs any convolutional coder for error correction. The conversion of every four binary data digits $\{s_i\}$ into one 16-point QAM signal q_i for transmission is here carried out by using a Gray code whose mapping is shown in Fig. 8.8. Thus, in Fig. 8.8, the two data digits s_{4i-3} and s_{4i-2} determine the quadrant in which q_i should lie, while the two data digits s_{4i-1} and s_{4i} determine the absolute values of the in-phase component a_i of q_i and the quadrature component b_i of q_i as follows

$$|a_i| = 2\,(s_{4i-1} + \tfrac{1}{2})/\sqrt{10} \qquad (8.23)$$

$$|b_i| = 2\,(s_{4i} + \tfrac{1}{2})/\sqrt{10} \qquad (8.24)$$

The uncoded system is otherwise the same as the coded system studied here.

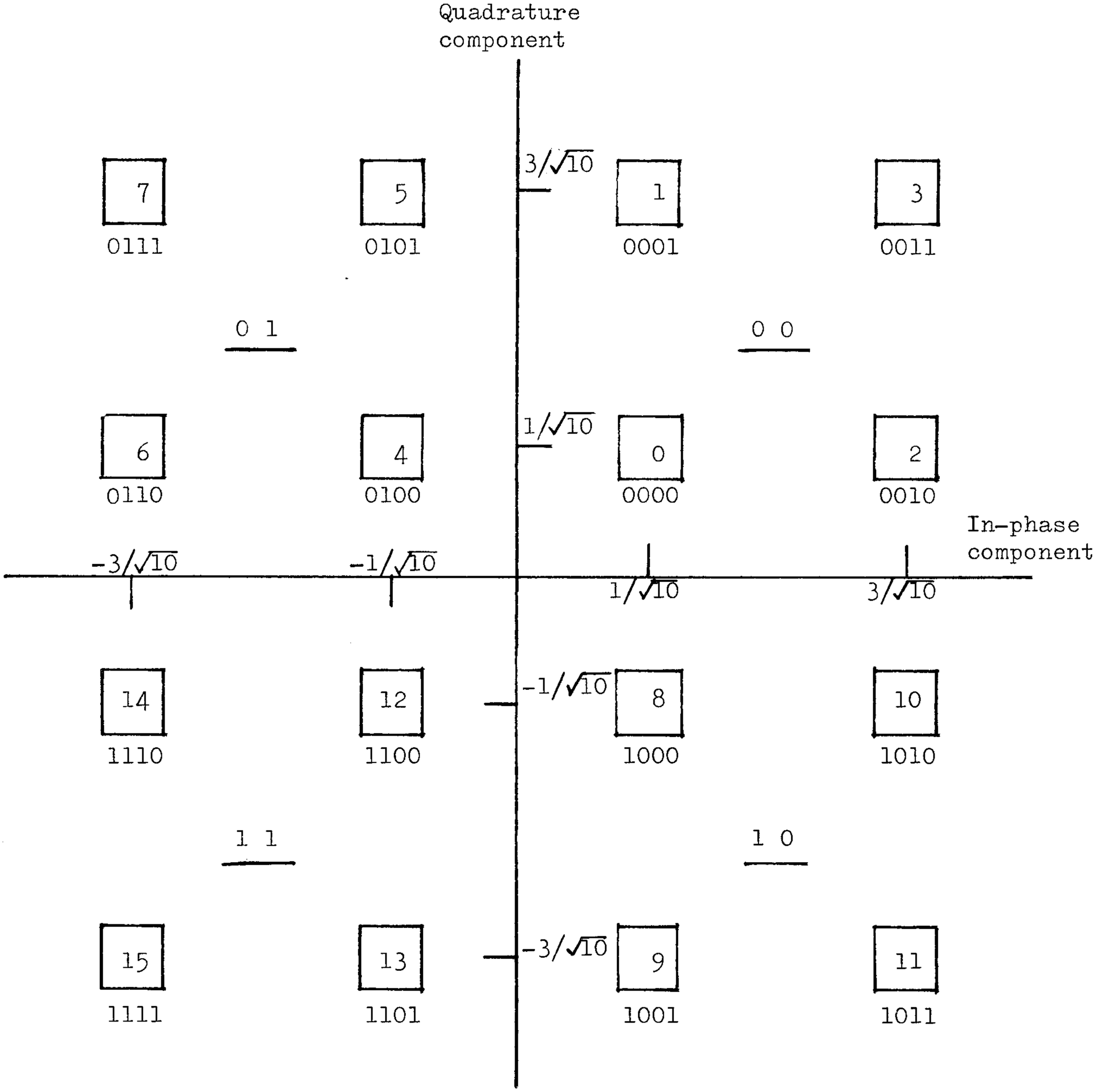


Fig. 8.8 Mapping of the 4 components s_{4i-3}, s_{4i-2}, s_{4i-1}, s_{4i} into the 16-point QAM signal q_i for the uncoded system.

8.8 Computer Simulation Results

Computer simulation tests have been carried out to determine the tolerances to Gaussian noise of the coded and the uncoded systems studied in this chapter. The sampled impulse responses of the four channels A, B, C, and D tested here are given in Table 6.4. The results of the tests are shown in Figs. 8.9 - 8.12 where the various systems considered here are summarised in Table 8.3. The 95% confidence limits of the results shown in Figs. 8.9 - 8.12 are about ± 0.5 dB. The signal to noise ratio here is defined as

$$\text{SNR} = 10 \log_{10}\left(\frac{E'}{\frac{1}{2}N_o} \right) \quad \text{dB} \qquad (8.25)$$

where E'is the average transmitted energy per signal element, and N_o is the two-sided power spectral density of the noise added at the output of the transmission path shown in Fig. 6.1.

All computer simulation tests have been carried out by using the CDC 7600 computer at Manchester, and the computer programs are written in FORTRAN.

Systems	m_e	Coding
A	0	uncoded
B1	10	Coder 1
B2	7	
B3	3	
C1	10	Coder 2
C2	7	
C3	3	
D1	10	Coder 3
D2	7	
D3	3	

Table 8.3 Systems tested (Figs. 8.9 - 8.12). m = 32, n = 16.
m_e : memory of the convolutional code
m : number of stored vectors $\{X_i\}$
4n : number of components in each vector X_i

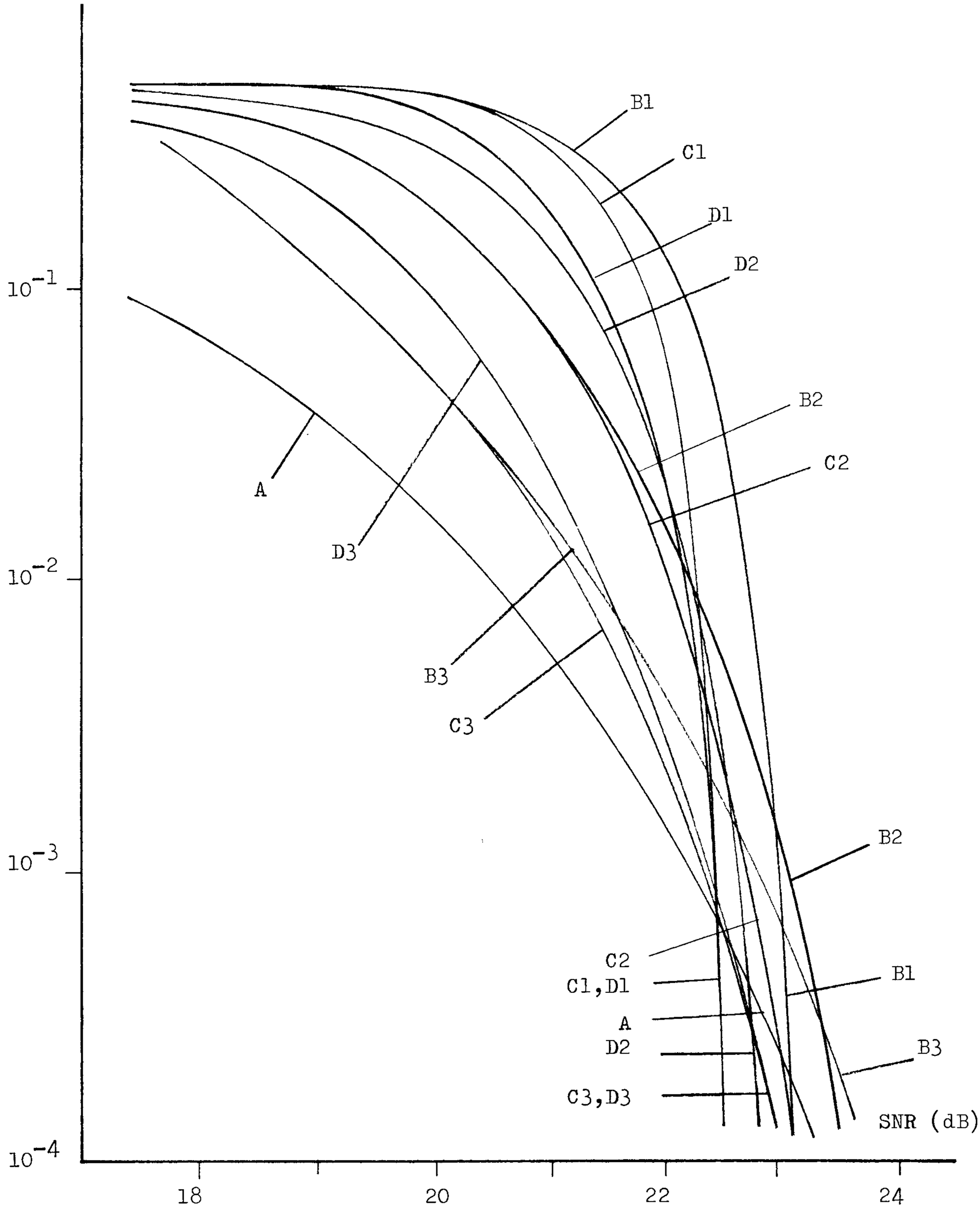


Fig. 8.9 Variation of error rate with signal to noise ratio for the various systems (Table 8.3) operating over channel A (Table 6.4).

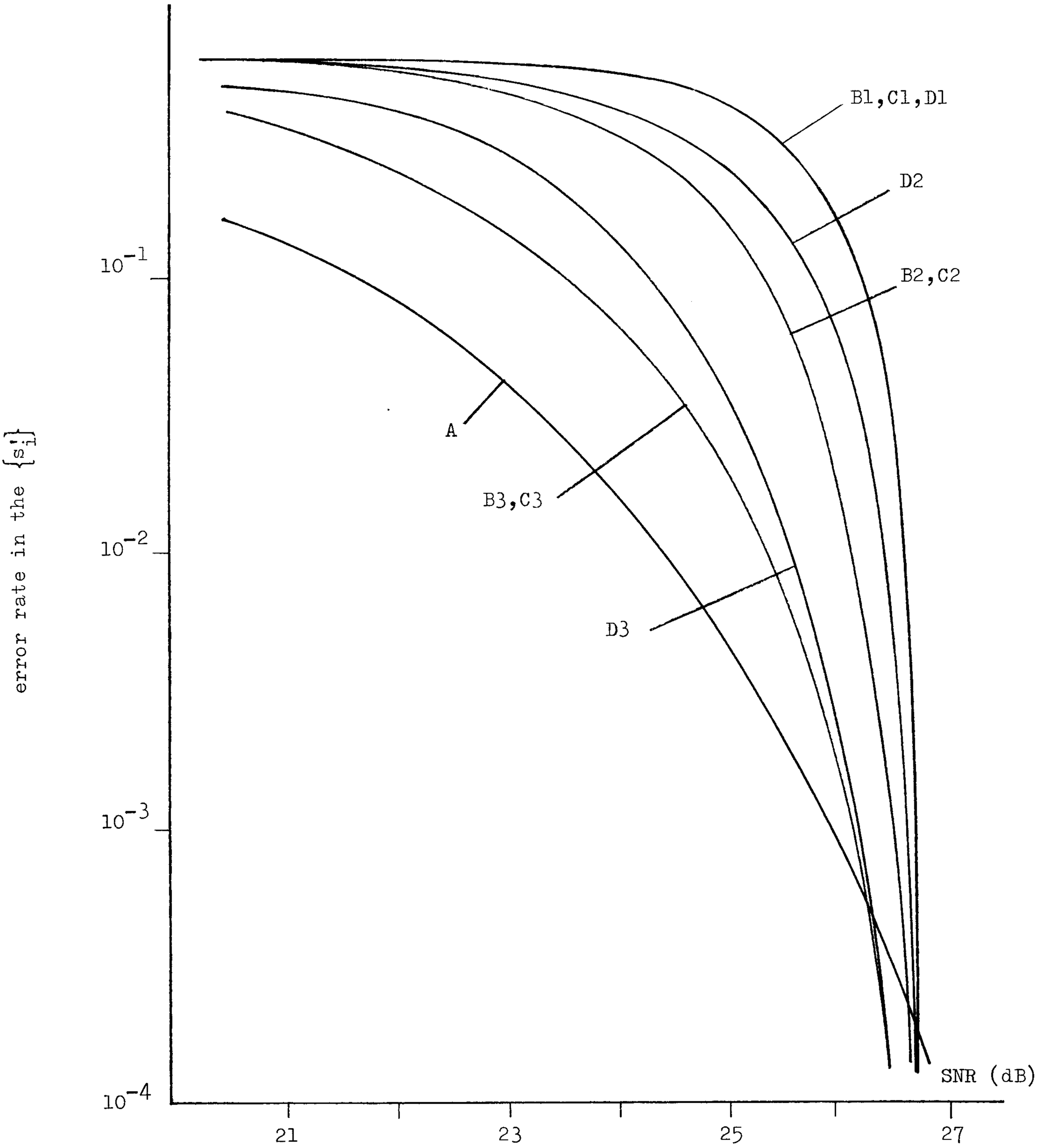


Fig. 8.10 Variation of error rate with signal to noise ratio for the various systems (Table 8.3) operating over channel B (Table 6.4).

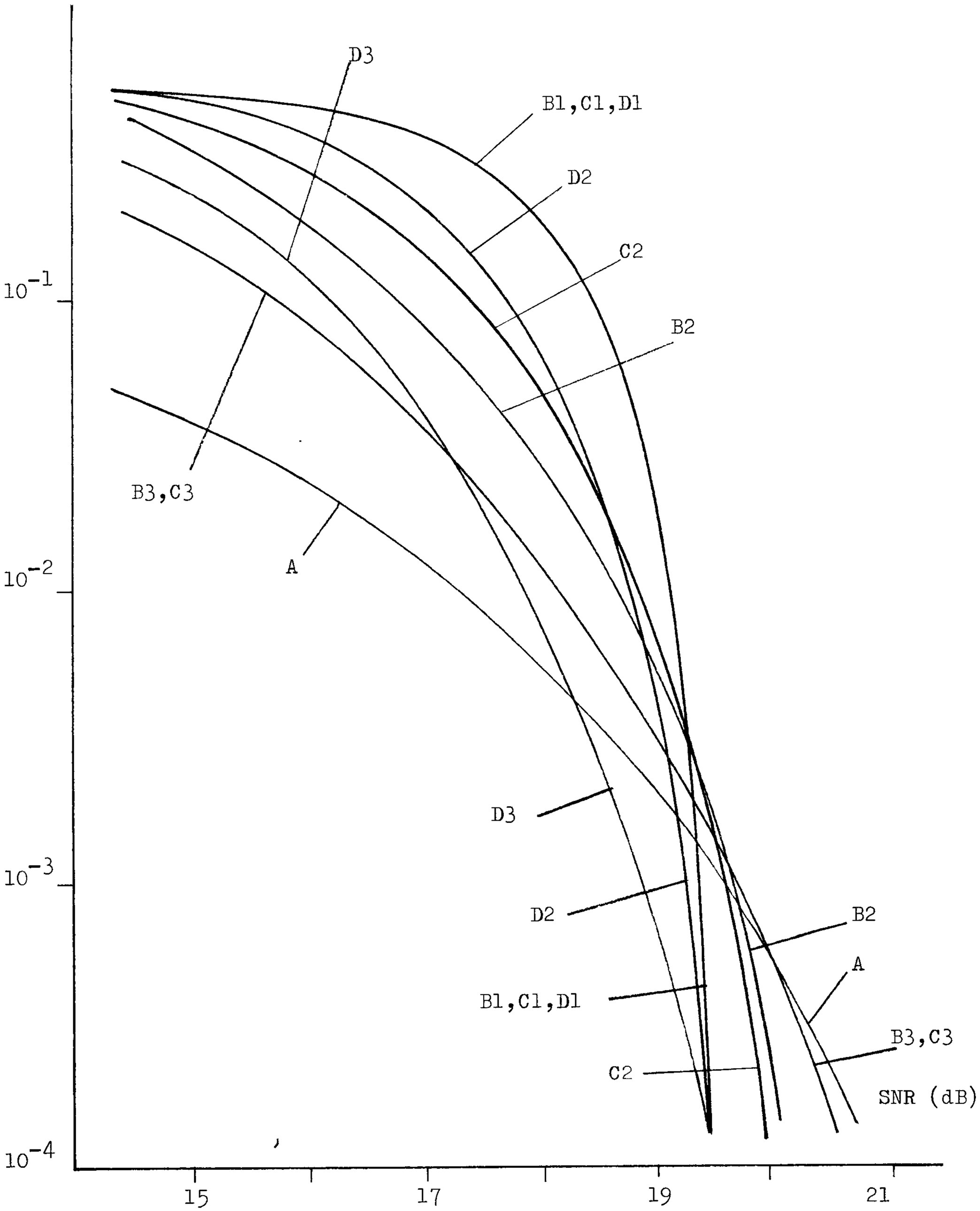


Fig. 8.11 Variation of error rate with signal to noise ratio for the various systems (Table 8.3) operating over channel C (Table 6.4).

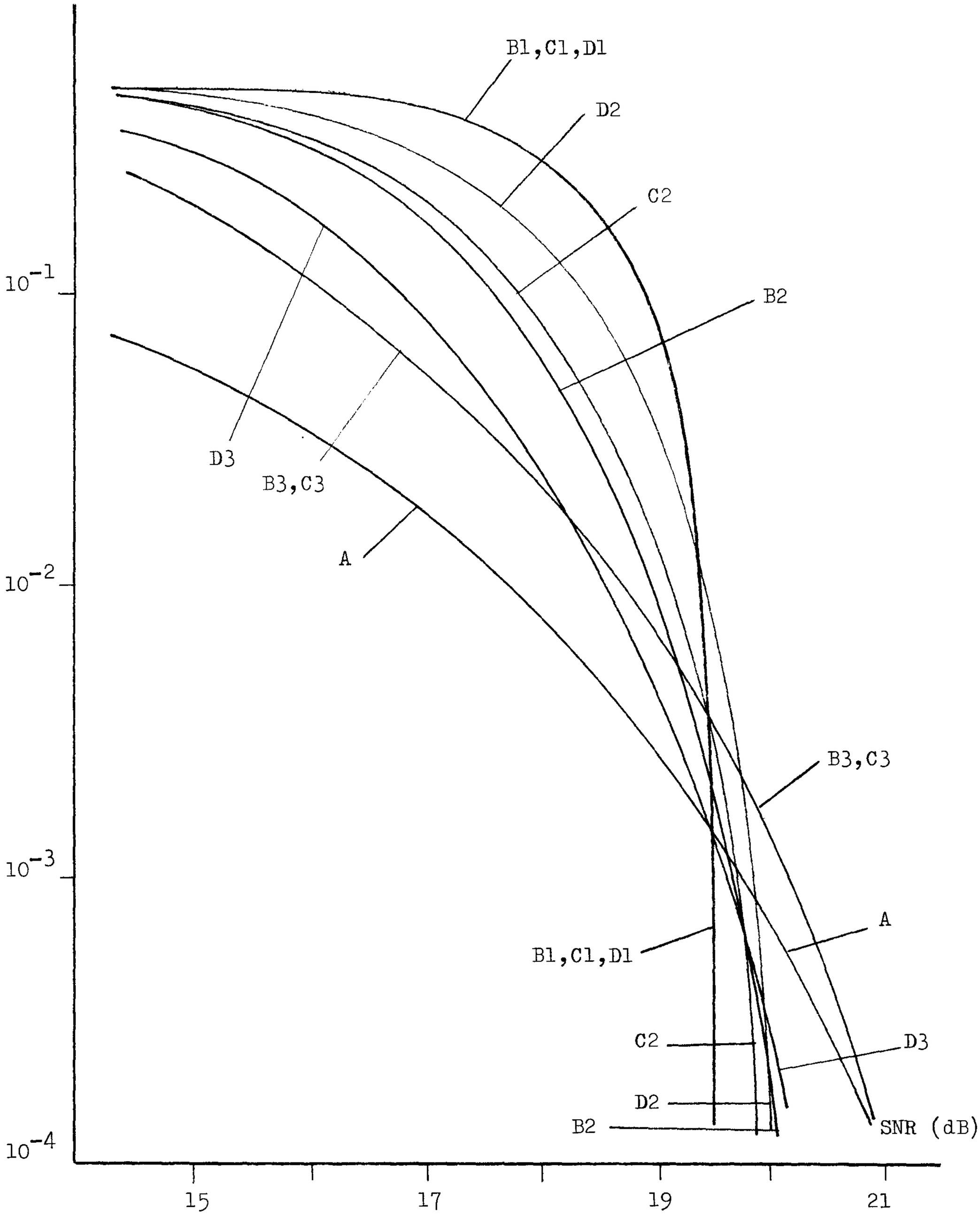


Fig. 8.12 Variation of error rate with signal to noise ratio for the various systems (Table 8.3) operating over channel D (Table 6.4).

8.9 Assessment of Systems

It can be seen from Figs. 8.9 - 8.12 and Table 8.3 that, for a given coding scheme (that is, coder 1, 2, or 3), the coded system that uses a larger value of m_e (memory of the convolutional code used in the system) generally yields a better tolerance to noise at low error rates while giving a more inferior tolerance to noise at high error rates. The coded system here also appears to have a better tolerance to noise than the uncoded system at sufficiently low error rates while having a poorer tolerance to noise than the uncoded system at high error rates. These results are in agreement with the results obtained in chapter 7 where a quaternary coded system and its corresponding binary uncoded system are studied.

Generally, for the four channels tested here, the coded system with coder 1 appears to have roughly the same tolerance to noise as that of the coded system with coder 2 although the latter is sometimes seen to have a slightly better tolerance to noise at low error rates, the same value of m_e being used in each of the two coders here. It therefore appears that the gain of coder 2 over coder 1 in having fewer possible values for the transmitted signals is approximately cancelled by the loss of coder 2 over coder 1 in having a reduced value of free distance for the code. Nevertheless, the coded system with coder 2 is likely to be more attractive than the coded system with coder 1 since it is likely to involve less equipment complexity by having a simpler coding process and a smaller number of possible values for the transmitted signals.

For a given value of m_e (Table 8.3), the coded system with coder 3 appears to have an inferior performance as compared to that of the coded system with coder 2 at high error rates. At low error rates,

however, the coded system with coder 3 tends to do better than the coded system with coder 2 here. This may, probably, be explained by assuming that the effective free distance of coder 2 is actually lower than that of coder 3 for the same value of m_e used in each of these coders, so that coder 2 effectively behaves like coder 3 but with a smaller effective m_e, bearing in mind that coders 2 and 3 both involve the coding of a binary data sequence into a 64-point QAM coded sequence.

The difference in performance between the use of different coders (that is, coders 1, 2, and 3) in the coded system here is, however, very marginal for the range of signal to noise ratios tested here. Coder 2 therefore appears to be the most attractive of the three coders, since it is likely to have the simplest practical implementation as can be seen from the descriptions of the various coders given in sections 8.3 - 8.5.

It can be seen from Figs. 8.9 - 8.12 that, the cross-over point at which there is no difference in the tolerance to noise between the uncoded system and the better coded system is generally at an error rate close to 10^{-3}, the coded system giving a better tolerance to noise than the uncoded system at lower error rates. At an error rate of 10^{-4}, the gain in tolerance to noise of the better coded system over the uncoded system is about 1 dB on each of the channels A and B, and is about 1.5 dB on each of the channels C and D. The value of m_e (Table 8.3) that gives the best performance at low error rates is 10 for any of the channels A, C, and D, and is 3 or 4 for channel B.

It has also been observed from the tests that, all the m (4n)-component stored vectors $\{X_i\}$ obtained at the end of the transmission in the coded system here have the same values for their corresponding components except for the last few components. This implies that, the value of

n used in the tests here (Table 8.3) is probably too large for the given value of m (=32) considered here. Consequently, a smaller value of n (that is, smaller than 16) may be used in the coded system with m = 32 here without introducing any reduction in the tolerance to noise of the system, bearing in mind that reducing the value of n also reduces the amount of storage and operations involved in the detection/decoding process of the system here.

The coded system developed in this chapter appears to be promising in those applications where a 16-point QAM signal is being transmitted at a rate close to the Nyquist rate of the given channel, and where it is required to reduce the error rate in the detected binary data values of the corresponding QAM signals to a value much lower than 10^{-4}.

CHAPTER 9

COMMENTS ON THE RESEARCH PROJECTS

9.1 Originality

All the experimental work described in this thesis which is not ascribed specifically to others, usually by quoting the appropriate references, is original to the best of the author's knowledge, although much of the new material has been originated either by Dr. A.P. Clark or through mutual discussions between Dr. A.P. Clark and the author. The followings are the more important contributions which have been originated by the author alone : -

Chapter 3 : Systems 3, 4, 5, 6, 7, 9, 10, and 11.

Chapter 4 : Systems 3, 5, and 6.

Chapter 5 : Systems 5 and 8.

Chapter 7 : Coder 2.

Chapter 8 : Coders 2 and 3.

All the computer simulation tests described in this thesis have been carried out by the author alone.

9.2 Conclusions

Various iterative detection processes have been developed and studied for use with a synchronous serial binary baseband data-transmission system. Among them, is the very promising detection process implemented as system 5 in chapter 3 which requires very few operations in achieving a near-optimum performance at high signal to noise ratios. The implementation of system 5 also appears to be relatively simple which makes it very attractive for use over a time-varying channel.

A systematic search detection process that involves a linear filtering of the received signals has been developed and studied for use with binary signals. The operation of this detection process is very simple and with the appropriate filter, such as that used in any of the systems 1, 2, 4, 5, and 6 in chapter 4, this detection process is able to achieve a near-optimum performance. The drawback of this detection process is that, a fairly large amount of storage is required in the system. The application of this detection process is restricted to be in the system that operates over a known time-invariant channel.

Near-maximum likelihood detection processes have also been considered for use with a 16-point QAM system operating over a severely distorted telephone channel. System 8 with version b or c in chapter 5 is the most promising of the various systems studied here. Its tolerance to Gaussian noise is generally better than that of the optimum nonlinear equalizer but is slightly inferior to that of system 1. The advantage of system 8 over system 1 is that, no linear prefilter is needed in the detection process, which reduces the synchronization period required in the system. Each of the systems 3, 4, and 5 uses a greatly shortened linear prefilter and is able to achieve only a small gain in tolerance to noise over the nonlinear equalizer at high signal to noise ratios.

Finally, the coded system that involves the use of a convolutional coder and a Gray coder at the transmitter, and a joint detection/decoding process at the receiver, has been developed and studied for use over a time dispersive channel. It is found that, this coded system has a better tolerance to Gaussian noise than the corresponding uncoded system at low error rates. The gain here appears to be larger when the channel introduces less severe amplitude or attenuation distortion.

Thus, for the uncoded data-transmission system considered in this thesis, a system that employs the maximum likelihood detection process is known to achieve the best tolerance to additive white Gaussian noise but it suffers from having an unduly complex implementation in practice. For systems operating with binary signals, system 3 of chapter 3 appears to be the simplest, in terms of equipment complexity, of all the more important systems developed here. System 5 of chapter 3 is, however, likely to be the most cost-effective system here. For systems operating with 16-point QAM signals, both system 1 and system 8 of chapter 5 appear to be very promising in terms of their performances in tolerance to Gaussian noise and the equipment complexity involved, as compared to the system employing the maximum likelihood detection process.

9.3 Possible Further Investigations

The iterative detection processes studied in chapter 3 have only been tested with binary signals. The more promising systems, such as systems 5, 6, 7, and 9, should be further tested for use with multi-level and QAM signals. Furthermore, methods of updating adaptively the tap-gain values of the filter used in system 9 may be studied and developed so as to make system 9 well suited for use over a time-varying channel.

The systematic search detection process developed in chapter 4 has the advantage of being able to operate with very high speed. However, the weakness of this detection process is that, a fairly large amount of storage is required in the system. Further investigations may therefore be carried out to modify the detection process here so as to reduce the amount of storage required in the system.

If the carrier phase of a data-transmission system is incorrectly detected at the receiver, then a long burst of errors in the detection of the transmitted data signal could occur. One usual way of overcoming this problem is to apply differential coding onto the system. Further investigations into applying the differential coding onto the more promising systems developed in chapters 5 and 8 would therefore be very useful.

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

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

REFERENCES

A GENERAL TOPICS IN COMMUNICATION THEORY

A1. Shannon,C.E., "A mathematical theory of communication", Bell Syst. Tech. J., 27, pp.379-423 and pp.623-656, July and October, 1948.

A2. Lee,Y.W., "Statistical theory of communication", John Wiley and Sons, Inc., New York, 1960.

A3. Wozencraft,J.M., Jacob,I.M., "Principles of communication engineering", John Wiley and Sons, Inc., New York, 1965.

A4. Lathi,B.P., "Random signals and communication theory", Intertext Student Editions, 1968.

A5. Van Trees,H.L., "Detection, estimation, and modulation theory, part-I", Willey, New York, 1968.

A6. Lucky,R.W., Salz,J., Weldon,E.J., "Principles of data communication", Mcgraw Hill, New York, 1968.

A7. Lucky,R.W., "A survey of the communication theory literature : 1968 - 1973", IEEE Trans. on Information Theory, IT-19, pp.725-739, 1973.

A8. Clark,A.P., "Principles of digital data transmission", Pentech Press, London, 1977.

A9. Clark,A.P., "Advance data-transmission systems", Pentech Press, London, 1977.

A10. Viterbi,A.J., Omura,J.K., "Principles of digital communication and coding", Mcgraw Hill, 1979.

A11. Schwartz,W., "Information transmission, modulation and noise", M graw Hill, New York, 1980.

B EQUALIZERS

B1. Rappeport,M.A., "Automatic equalization of data transmission facility distortion using transversal equalizers", IEEE Trans. on Communication Technology, COM-12, pp.65-73, September 1964.

B2. Becker,F.K., Holzman,L.N., Lucky,R.W., Port,E., "Automatical equalization for digital communication", Proc. IEEE, 53, pp.96-97, January 1965.

B3. Lucky,R.W., "Automatic equalization for digital communication", Bell Syst. Tech. J., 44, pp.547-588, April 1965.

B4. Lucky,R.W., "Techniques for adaptive equalization of digital communication systems", Bell Syst. Tech. J., 45, pp.255-286, Febrary 1966.

B5. Lucky,R.W., Rudin,H.R., "An automatic equalizer for general purpose communication channels", Bell Syst. Tech. J., 46, pp.2179-2208, November 1967.

B6. Lytle,D.W., "Convergence criteria for transversal equalizers", Bell Syst. Tech. J., 47, pp.1775-1800, October 1968.

B7. Lucky,R.W., Salz,J., Weldon,E.J., "Principles of data communication", Mcgraw Hill, New York, 1968.

B8. Gersho,A., "Adaptive equalization of highly dispersive channels for data transmission", Bell Syst. Tech. J., 48, pp.55-70, January 1969.

B9. Proakis,J.G., Miller,J.H., "An adaptive receiver for digital signalling through channels with intersymbol interference", IEEE Trans. on Information Theory, IT-15, pp.484-497, July 1969.

B10. Rudin,H.R., "A continuously adaptive equalizer for general purpose communication channels", Bell Syst. Tech. J., 48, pp.1865-1884, July-August 1969.

B11. Proakis,J.G., "Adaptive digital filters for equalization of telephone channels", IEEE Trans. on Audio and Electroacoustics, AU-18, pp.195-200, June 1970.

B12. Hirsh,D., Wolf,W.J., "A simple adaptive equalizer for efficient data transmission",
IEEE Trans. on Communication Technology, COM-18, pp.5-12,
February 1970.

B13. Lender,A., "Decision-directed digital adaptive equalization technique for high speed data transmission",
IEEE Trans. on Communication Technology, COM-18, pp.625-632.,
October 1970.

B14. Newhall,E.E., Qureshi,S.U.H., Simone,C.F., "A technique for finding approximate inverse systems and its application to equalization",
IEEE Trans. on Communication Technology, COM-19, pp.1116-1127,
December 1971.

B15. Mark,J.W., Haykin,S.S., "Adaptive equalization for digital communication",
Proc. IEE, 118, pp.1711-1720, December 1971.

B16. Monsen,P., "Feedback equalization for fading dispersive channels",
IEEE Trans. on Information Theory, IT-17, pp.56-64, January 1971.

B17. Tomlinson,M., "New automatic equalizer employing modulo arithmetric",
Electronics Letters, 7, pp.138-139, 25, March 1971.

B18. Taylor,D.P., "Nonlinear feedback equalizer employing a soft limiter",
Electronics Letters, 7, pp.265-267, 20, May 1971.

B19. George,D.A., Bowen,R.R., Storey,J.R., "An adaptive decision feedback equalizer",
IEEE Trans. on Communication Technology, COM-19, pp.281-293,
June 1971.

B20. Ungerboeck,G., "Nonlinear equalization of binary signals in Gaussian noise",
IEEE Trans. on Communication Technology, COM-19, pp.1128-1137,
December 1971.

B21. Bershad,N.J., Vena,P.A., "Eliminating intersymbol interference - a state space approach",
IEEE Trans. on Information Theory, IT-18, pp.275-281, March 1972.

B22. Westcott,R.J., "An experimental adaptive equalized modem for data transmission over the switched telephone network", Radio and Electron Eng., 42, pp.499-507, November 1972.

B23. Mark,J.W., Budihardjo,P.S., "Joint optimization of receiver filter and equalizer", IEEE Trans. on Communication Technology, COM-21, pp.264-267, March 1973.

B24. Qureshi,S.U.H., "Adjustment of the position of the reference tap of an adaptive equalizer", IEEE Trans. on Communication Technology, COM-21, pp.1046-1052, September 1972.

B25. Clark,A.P., "Design technique for nonlinear equalizers", Proc. IEE, 120, pp.329-333, March 1973.

B26. Mark,J.W., Budihardjo,P.S., "Performance of jointly optimised prefilter-equalizer receivers", IEEE Trans. on Communication Technology, COM-21, pp.941-945, August 1973.

B27. Taylor,D.P., "The estimate feedback equalizer : a suboptimum nonlinear receiver", IEEE Trans. on Communication Technology, COM-21, pp.979-990, September 1973.

B28. Kobayashi,H., Tang,D.T., "A decision feedback receiver for channels with strong intersymbol interference", IBM J. Res. Developm., 17, pp.413-419, September 1973.

B29. Salz,J., "Optimum mean-square decision feedback equalization", Bell Syst. Tech. J., 52, pp.1341-1373, October 1973.

B30. Masserschmitt,D.G., "A geometric theory of intersymbol interference, part 1", Bell Syst. Tech. J., 52, pp.1483-1519, November 1973.

B31. Falconer,D.D., Foshini,G.J., "Theory of minimum mean square error QAM systems employing decision feedback equalization", Bell Syst. Tech. J., 52, pp.1821-1849, December 1973.

B32. Salazar,A.C., "Design of transmitter and receiver filters for decision feedback equalization", Bell Syst. Tech. J., 53, pp.503-523, March 1974.

B33. Macleed,C.J., "Study of recursive equalizers for data transmission with a comparison of the performance of six systems", Proc. IEE, 122, pp.1097-1104, October 1975.

B34. Clark,A.P., Tint,U.S., "Linear and nonlinear equalizers for baseband channels", Radio and Electronic Eng., 45, pp.271-283, June 1975.

B35. Clark,A.P., Serinken,M.N., "Nonlinear equalizer using modulo arithmetic", Proc. IEE, 123, pp.32-38, January 1976.

B36. Clark,A.P., "Advance data-transmission systems", Pentech Press, London, 1977.

B37. Kawas-kaleh,G., "Double decision feedback equalizer", Frequenz, 33, pp.146-149, 1979.

C DETECTION PROCESSES

C1. Chang,R.W., Hancock,J.C., "On receiver structures for channels having memory",
IEEE Trans. on Information Theory, IT-12, pp.463-468, October 1966.

C2. Gonsalves, R.A., "Maximum likelihood receiver for digital data transmission",
IEEE Trans. on Communication Technology, COM-16, pp.392-398, June 1968.

C3. Abend,K., Harley,T.J., Fritchman,B.D., Gumacos,C., "On optimum receivers for channels having memory",
IEEE Trans. on Information Theory, IT-14, pp.819-820, November 1968.

C4. Bowen,R.R., "Bayesian decision procedure for interfering digital signals",
IEEE Trans. on Information Theory, IT-15, pp.506-507, July 1969.

C5. Clark,A.P., "The transmission of digitally-coded-speech signals by means of random access discrete address systems",
Ph.D. thesis, Imperial College, London 1969.

C6. Abend,K., Fritchman,B.D., "Statistical detection for communication channels with intersymbol interference",
Proc. IEEE, 58, pp.779-785, May 1970.

C7. Clark,A.P., "Adaptive detection of distorted digital signals",
Radio and Electronic Eng., 40, pp.107-119, September 1970.

C8. Clark,A.P., "A synchronous serial data-transmission system using orthogonal groups of binary signal elements",
IEEE Trans. on Communication Technology, COM-19, pp.1101-1110, December 1971.

C9. Clark,A.P., "Adaptive detection with intersymbol interference cancellation for distorted digital signals",
IEEE Trans. on Communication Technology, COM-20, pp.350-361, June 1972.

C10. Forney,G.D., "Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference",
IEEE Trans. on Information Theory, IT-18, pp.363-378, May 1972.

C11. Magee,F.R., Proakis,J.G., "Maximum likelihood sequence estimation for digital signalling in the presence of intersymbol interference", IEEE Trans. on Information Theory, IT-19, pp.120-124, January 1973.

C12. Forney,G.D., "The Viterbi algorithm", Proc. IEEE, 61, pp.268-278, March 1973.

C13. Qureshi,S.U.H., Newhall,E.E., "An adaptive receiver for data transmission over time-dispersive channels", IEEE Trans. on Information Theory, IT-19, pp.468-457, July 1973.

C14. Clark,A.P., Ghani,F., "Detection processes for orthogonal groups of digital signals", IEEE Trans. on Communication Technology, COM-21, pp.907-915. August 1973.

C15. Messerschmitt,D.G., "A geometric theory of intersymbol interference, part II", Bell Syst. Tech. J., 52, pp.1521-1539, November 1973.

C16. Falconer,D.D., Magee,F.R., "Adaptive channel memory truncation for maximum likelihood sequence estimation", Bell Syst. Tech. J., 52, pp.1541-1562, November 1973.

C17. Pulleyblank,R.W., "A comparison of receivers designed on the basis of minimum mean-square error and probability of error for channels with intersymbol interference and noise", IEEE Trans. on Communication Technology, COM-21, pp.1434-1438, December 1973.

C18. Fritchman,B.D., Kanal,L.N., Womer,J.D., "Optimum sequential detector performance on intersymbol interference channels", IEEE Trans. on Communication Technology, COM-22, pp.788-797, June 1974.

C19. Fredricsson,S.A., "Optimum transmitting filter in digital PAM systems with a Viterbi detector", IEEE Trans. on Information Theory, IT-20, pp.479-489, July 1974.

C20. Cantoni,A., Kwong,K., "Further results on the Viterbi algorithm equalizer", IEEE Trans. on Information Theory, IT-20, pp.764-767, November 1974.

C21. Foschini,J.G., "Performance bound for maximum likelihood reception of digital data",
IEEE Trans. on Information Theory, IT-21, pp.47-50, January 1975.

C22. Magee,F.R., "A comparison of compromise Viterbi algorithm and standard equalization techniques over band-limited channels",
IEEE Trans. on Communication Technology, COM-23, pp.361-367, March 1975.

C23. Hayes,J.F., "The Viterbi algorithm applied to digital data transmission",
Commun. Soc., 13, pp.15-20, January 1975.

C24. Yao,K., Milstein,L.B., "On maximum likelihood bit detection of binary signals with intersymbol interference in Gaussian noise",
IEEE Trans. on Communication Technology, COM-23, pp.971-976, September 1975.

C25. Gitlin,G.D., Ho,E.Y., "A null zone decision feedback equalizer incorporating maximum likelihood bit detection",
IEEE Trans. on Communication Technology, COM-23, pp.1243-1250, November 1975.

C26. Fredricsson,S.A., "Joint optimization of transmitter and receiver filters in digital PAM systems with a Viterbi detector",
IEEE Trans. on Information Theory, IT-22, pp.200-210, March 1976.

C27. Falconer,D.D., Magee,F.R., "Evaluation of decision feedback equalization and Viterbi algorithm detection for voiceband data transmission, part I and part II",
IEEE Trans. on Communication Technology, COM-24, pp.1130-1139, October 1976; pp.1238-1245, November 1976.

C28. Clements,A., "The application of iterative techniques to adaptive detection processes",
Ph.D. thesis, Loughborough University of Technology, May 1976.

C29. Clark,A.P., Harvey,J.D., "Detection processes for distorted binary signals",
Radio and Electronic Eng., 46, pp.533-542, November 1976.

C30. Clark,A.P., Harvey,J.D., "Detection of distorted QAM signals",
IEE Journal on E.C.S., pp.103-109, April 1977.

C31. Clark,A.P., Harvey,J.D., Driscoll,J.P., "Improved detection processes for distorted digital signals",
IERE Conference on digital processing of signals in Communications, University of Technology, Loughborough, 1977.

C32. Clark,A.P., "Advance data-transmission systems",
Pentech Press, London, 1977.

C33. Desblache,A.E., "Optimal short desired impulse response for maximum likelihood sequence estimation",
IEEE Trans. on Communication Technology, COM-25, pp.735-738, July 1977.

C34. Lee,W.U., Hill,F.S., "A maximum likelihood sequence estimator with decision feedback equalization",
IEEE Trans. on Communication Technology, COM-25, pp.971-979, 1977.

C35. Foschini,G.J., "A reduced state variant of maximum likelihood sequence detection attaining optimum performance for high signal to noise ratio",
IEEE Trans. on Information Theory, IT-23, pp.605-609, September 1977.

C36. Harvey,J.D., "Adaptive detection of digital suppressed carrier A.M. signals",
Ph.D. thesis, Loughborough University of Technology, February 1978.

C37. Clark,A.P., Harvey,J.D., Driscoll,J.P., "Near-maximum likelihood detection processes for distorted digital signals",
Radio and Electronic Eng., 48, pp.301-309, June 1978.

C38. Clark,A.P., Kwong,C.P., Harvey,J.D., "Detection processes for severely distorted digital signals",
Electronic Circuits and Systems, 3, pp.27-37, January 1979.

C39. Clark,A.P., Kwong,C.P., Mcverry,F., "Estimation of the sampled impulse response of a channel",
Signal Processing, 2, pp.39-53, January 1980.

C40. Viterbi,A.J., Omura,J.K., "Principles of digital communication and coding",
Mcgraw Hill, 1979.

C41. Clark,A.P., Asghar,S.M., "Detection of digital signals with known time-varying intersymbol interference",
IEE Proc., F, 128, pp.167-174, June 1981.

C42. Clark,A.P., Fairfield,M.J., "Detection processes for a 9600 bit/s modem",
IERE, 51, pp.455-465, September 1981.

C43. Clark,A.P., Mcverry,F., "Channel estimation for an HF radio link",
IEE Proc., 128, F, pp.33-42, February 1981.

D CODING AND DECODING PROCESSES ON CONVOLUTIONAL CODES

D1. Wozencraft,J.M., Reiffen,B., "Sequential decoding",
Cambridge, Mass. : M.I.T. Press, 1961.

D2. Massey,J.L., "Threshold decoding",
Cambridge, Mass. : M.I.T. Press, 1963.

D3. Fano,R.M., "A heuristic discussion of probabilistic decoding",
IEEE Trans. on Information Theory, IT-9, pp.64-74, April 1963.

D4. Forney,G.D., "Generalised minimum distance decoding",
IEEE Trans. on Information Theory, IT-12, pp.125-131, April 1966.

D5. Viterbi,A.J., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm",
IEEE Trans. on Information Theory, IT-13, pp.260-269, April 1967.

D6. Jacobs,I.M., "Sequential decoding for efficient communication from deep space",
IEEE Trans. on Communication Technology, COM-15, pp.492-501, August 1967.

D7. Kohlenberg,A., Forney,G.D., "Convolutional coding for channels with memory",
IEEE Trans. on Information Theory, IT-14, pp.618-626, September 1968.

D8. Omura,J.K., "On the Viterbi decoding algorithm",
IEEE Trans. on Information Theory, IT-15, pp.177-179, January 1969.

D9. Jelinek,F., "Fast sequential decoding algorithm using a stack",
IBM J. Res. Dev., 13, pp.675-685, November 1969.

D10. Jelinek,F., Bahl,L.R., "Maximum likelihood and sequential decoding of short constraint length convolutional codes",
Proc. 7th Annu. Allerton Conference, pp.130-139, October 1969.

D11. Costello,D.J., "A construction technique for random-error-correcting convolutional codes",
IEEE Trans. on Information Theory (Corresp.), IT-15, pp.631-636, September 1969.

D12. Forney,G.D., "Convolutional codes I : Algebraic structure",
IEEE Trans. on Information Theory, IT-16, pp.720-738, November 1970.

D13. Forney,G.D., "Coding and its application in space communications", IEEE Spectrum, 7, pp.47-58, June 1970.

D14. Massey,J.L., Constello,D.J., "Nonsystematic convolutional codes for sequential decoding in space applications", IEEE Trans. on Communication Technology, COM-19, pp.806-813, October 1971.

D15. Heller,J.A., Jacobs,I.M., "Viterbi decoding for satellite and space communication", IEEE Trans. on Communication Technology, COM-19, pp.835-848, October 1971.

D16. Viterbi,A.J., "Convolutional codes and their performance in communication systems", IEEE Trans. on Communication Technology, COM-19, pp.751-772, October 1971.

D17. Forney,G.D., Bower,E.K., "A high speed sequential decoder prototype design and test", IEEE Trans. on Communication Technology, COM-19, pp.821-835, October 1971.

D18. Omura,J.K., "Optimal receiver design for convolutional codes and channels with memory via control theoretical concepts", Inform. Sci., 3, pp.243-266, July 1971.

D19. Kobayashi,H., "Correlative level coding and maximum likelihood decoding", IEEE Trans. on Information Theory, IT-17, pp.586-594, September 1971.

D20. Bahl,L.R., Cullum,C.D., Frazer,W.D., Jelinek,F., "An efficient algorithm for computing free distance", IEEE Trans. on Information Theory (Corresp.), IT-18, pp.437-439, May 1972.

D21. Larsen,K.J., "Comments on 'An efficient algorithm for computing free distance'", IEEE Trans. on Information Theory (Corresp.), IT-19, pp.577-579, July 1973.

D22. Helgert,H.J., Stinaff,R.D., "Minimum-distance bounds for binary linear codes", IEEE Trans. on Information Theory, IT-19, pp.344-356, May 1973.

D23. Forney,G.D., "Structural analysis of convolutional codes via dual codes",
IEEE Trans. on Information Theory, IT-19, pp.512-518, July 1973.

D24. Forney,G.D., "The Viterbi Algorithm",
Proc. IEEE, 61, pp.268-278, March 1973.

D25. Chase,D., "A combined coding and modulation approach for communication over dispersive channels",
IEEE Trans. on Communication Technology, CCM-21, pp.159-174,
March 1973.

D26. Mecklenburg,P., Pehlert,W.K., Sullivan,D.D., "Correction of errors in multilevel Gray-coded data",
IEEE Trans. on Information Theory, IT-19, pp.336-340, May 1973.

D27. Paaske,E., "Short binary convolutional codes with maximal free distance for rates $\frac{2}{3}$ and $\frac{3}{4}$",
IEEE Trans. on Information Theory, IT-20, pp.683-688, September 1974.

D28. Anderson,J.B., "A stack algorithm for source coding with a fidelity criterion",
IEEE Trans. on Information Theory, IT-20, pp.211-226, May 1974.

D29. Haccoun,D., Ferguson,M.J., "Generalized stack algorithms for decoding convolutional codes",
IEEE Trans. on Information Theory, IT-21, pp.638-651, November 1975.

D30. Johannesson,R., "Robustly optimum rate one-half binary convolutional codes",
IEEE Trans. on Information Theory, IT-21, pp.464-468, July 1975.

D31. Chevillat,P.R., Costello,D.J., "Distance and computation in sequential decoding",
IEEE Trans. on Communication Technology, COM-24, pp.440-447,
April 1976.

D32. Ng,W.H., Kim,F., Tashiro,S., "Maximum likelihood decoding scheme for convolutional codes",
International Telemetering Conference paper, 12, pp.154-164,
September 1976.

D33. Ungerboeck,G., "Channel coding with multi-level phase signals",
IEEE International Symposium on Information Theory, Sweden,
June 1976.

D34. Davies,B.H., Foley,G., "The implementation of Viterbi decoding on satellite communication circuits",
IERE Conference Proceedings 37, pp.159-172, 1977.

D35. Digeon,A., "On improving bit error probability of QPSK and 4-level amplitude modulation systems by convolutional coding",
IEEE Trans. on Communication Technology, COM-25, pp.1236,
October 1977.

D36. Imai,H., Hirakawa,S., "A new multilevel coding method using error correcting codes",
IEEE Trans. on Information Theory, IT-23, pp.371-377, 1977.

D37. Hemmati,F., Costello,D.J., "Truncated error probability in Viterbi decoding",
IEEE Trans. on Communication Technology, COM-25, pp.530-532,
May 1977.

D38. Chevillat,P.R., Costello,D.J., "A multiple stack algorithm for erasurefree decoding of convolutional codes",
IEEE Trans. on Communication Technology, COM-25, pp.1460-1470,
December 1977.

D39. Fritchman,B.D., Hafer,C.D., Mixsell,J.C., Sattar,M.A., "approximation to a joint detection/decoding algorithm",
IEEE Trans. on Communication Technology, COM-25, pp.271-278,
February 1977.

D40. Ng.W.H., Goodman,R.M.F., "An efficient minimum-distance decoding algorithm for convolutional error correcting codes",
Proc. IEE, 125, pp.97-103, February 1978.

D41. Richer,I.R.A., "A simple interleaver for use with Viterbi decoding",
IEEE Trans. on Communication Technology, COM-26, pp.406-408,
March 1978.

D42. Clark,A.P., "Minimum distance decoding of binary convolutional codes",
Computers and Digital Techniques, 1, pp.190-196, October 1978.

D43. Acampora,A.S., "Maximum likelihood decoding of binary convolutional codes",
IEEE Trans. on Communication Technology, COM-26, pp.766-776,
June 1978.

D44. Rashvand,H.F., Farrell,P.G., "A new approach to finding the free-distance and weight distribution of a convolutional code", IEEE International Symposium on Information Theory, Grignano, Italy 1979.

D45. Farrell,P.G., Tait,D.J., Borelli,W.C., "A convolutional code using soft decoding decision for practical line transmission", IEE Colloquium on data transmission codes, London, November 1980.

D46. Borelli,W.C., Rashvand,H.F., Farrell,P.G., "Convolutional codes for multilevel modems", IEE Electronic Letters, 17, pp.331-333, 30th April 1981.

D47. Clark,A.P., Ser,W., "Improvement in tolerance to noise through the transmission of multilevel coded signals", IERE Conference on digital processing of signals in Communications, University of Technology, Loughborough, England, 7-10 April 1981, pp. 129-141.

D48. Rashvand,H.F., "Effective minimum free distance for convolutional codes", Electronic Letters, 17, pp.463-465, 25th May 1981.

D49. Goodman,R.M.F., Winfield,A.F.T., "Soft-decision minimum-distance sequential decoding algorithm for convolutional codes", IEE Proc. F, 128, pp.179-186, June 1981.

E MATHEMATICAL TOPICS

E1. Varge,R.S., "Matrix iterative analysis",
Prentice Hall, 1962.

E2. Russel,D., "Optimization theory",
W.A.Benjamin Inc., New York, 1970.

E3. Adby,P.R., Dempster,M.A.H., "introduction to optimization methods",
Chapman and Hall, London, 1974.

E4. Clark,A.P., "Vectors and Matrices",
Lecture notes, Dept. of Electronic and Electrical Engineering,
Loughborough University.

E5. Hestenes,H.R., "Optimization theory",
John Wiley and Sons, Inc., 1975.

APPENDIX A1

DERIVATION OF THE THRESHOLD VALUES THAT PREVENT THE ITERATIVE PROCESSES OF SYSTEMS 1 - 7 (CHAPTER 3) FROM DIVERGING

This appendix shows that, the iterative process used in any of the systems 1 - 7 (chapter 3) is prevented from diverging if the threshold value t_h (Tables 3.1 and 3.2) used in the updating process for the hth component x_h of the n-component vector X (eqn. 3.13) is such that

$$t_h \geq \|Y_h\|^2 \quad , \quad \text{if} \quad x_h = \pm 1$$
$$t_h \geq \tfrac{1}{2}\|Y_h\|^2 \quad , \quad \text{if} \quad x_h = 0 \qquad \text{(A1.1)}$$

for $h = 1, 2, \cdots, n$ where the value of x_h here is referred to its value before it is updated, and $\|Y_h\|$ is the Euclidean norm or length of the n-component vector Y_h (eqn. 3.9). The iterative process here employs the arrangement of Fig. 3.4, and divergence is defined to have occurred when the value of $\|R_a\|$ increases during the iterative process, where R_a is the n-component vector defined by eqn. 3.22.

Consider first the case when $x_h = \pm 1$. In the updating process for x_h, the quantity e_h (eqn. 3.24) is first evaluated at the output of the network Y_h^T shown in Fig. 3.4 as

$$e_h = R_a Y_h^T \qquad \text{(A1.2)}$$

where Y_h^T is the transpose of Y_h. The increment Δx_h of x_h is next determined using Table 3.2. Thus, if the value of x_h has the same sign as that of e_h, or if $|e_h| \leq t_h$, then the value of Δx_h is set to zero and the

vector R_a (being updated by using eqn. 3.31) remains unchanged so that neither divergence nor convergence can occur here. If, however, the value of x_h has a different sign from that of e_h and if

$$|e_h| > t_h \qquad \text{(A1.3)}$$

then the value of Δx_h is set to 2 or -2 depending on whether x_h is negative or positive. The vector R_a is now updated as (eqn. 3.31)

$$(R_a)_{new} = R_a \pm 2Y_h \qquad \text{(A1.4)}$$

In the n-dimensional Euclidean vector space containning the vectors R_a and Y_h, the quantity e_h / Y_h (eqn. A1.2) is the orthogonal projection of R_a onto Y_h, as shown in Fig. A1.1. In Fig. A1.1, OA represents the vector of R_a before it is updated, and OB represents the updated vector of R_a. The Euclidean norm $\|R_a\|$ is here given by the corresponding value of the length of OA or OB. The straight line AB here is parallel to the direction of the vector Y_h as is implied by eqn. A1.4. It can be seen from Fig. A1.1 that, if B lies between A and C, then the length of OB is smaller than that of OA, so that the Euclidean norm $\|R_a\|$ for the updated vector of R_a is now smaller than that for the vector of R_a before it is updated, and hence divergence is prevented from occurring here. Thus, from Fig. A1.1, B lies between A and C if

$$\frac{2|e_h|}{\|Y_h\|} > 2\|Y_h\|$$

or

$$|e_h| > \|Y_h\|^2 \qquad \text{(A1.5)}$$

Since $|e_h| > t_h$ here, therefore the inequality of (A1.5) is always

satisfied so long as

$$t_h \geq \|Y_h\|^2 \qquad \text{(A1.6)}$$

which is therefore the necessary condition for the iterative process to be prevented from diverging when the value of x_h is 1 or -1.

Consider now the case when $x_h = 0$. It can be seen from Table 3.1 that, if $|e_h| \leq t_h$, then Δx_h is set to zero and the vector R_a (being updated by using eqn. 3.31) is unchanged so that neither divergence nor convergence can occur here. If, however, $|e_h| > t_h$, then Δx_h is set to 1 or -1 depending on whether e_h is positive or negative, and R_a is now updated as

$$(R_a)_{new} = R_a \pm Y_h \qquad \text{(A1.7)}$$

which obviously resembles eqn. A1.4. The same argument used for the case when $x_h = \pm 1$ can thus be applied here to show that the necessary condition for the iterative process to be prevented from diverging, when the value of x_h is zero, is

$$t_h \geq \tfrac{1}{2}\|Y_h\|^2 \qquad \text{(A1.8)}$$

which is similar to that given by (A1.5)

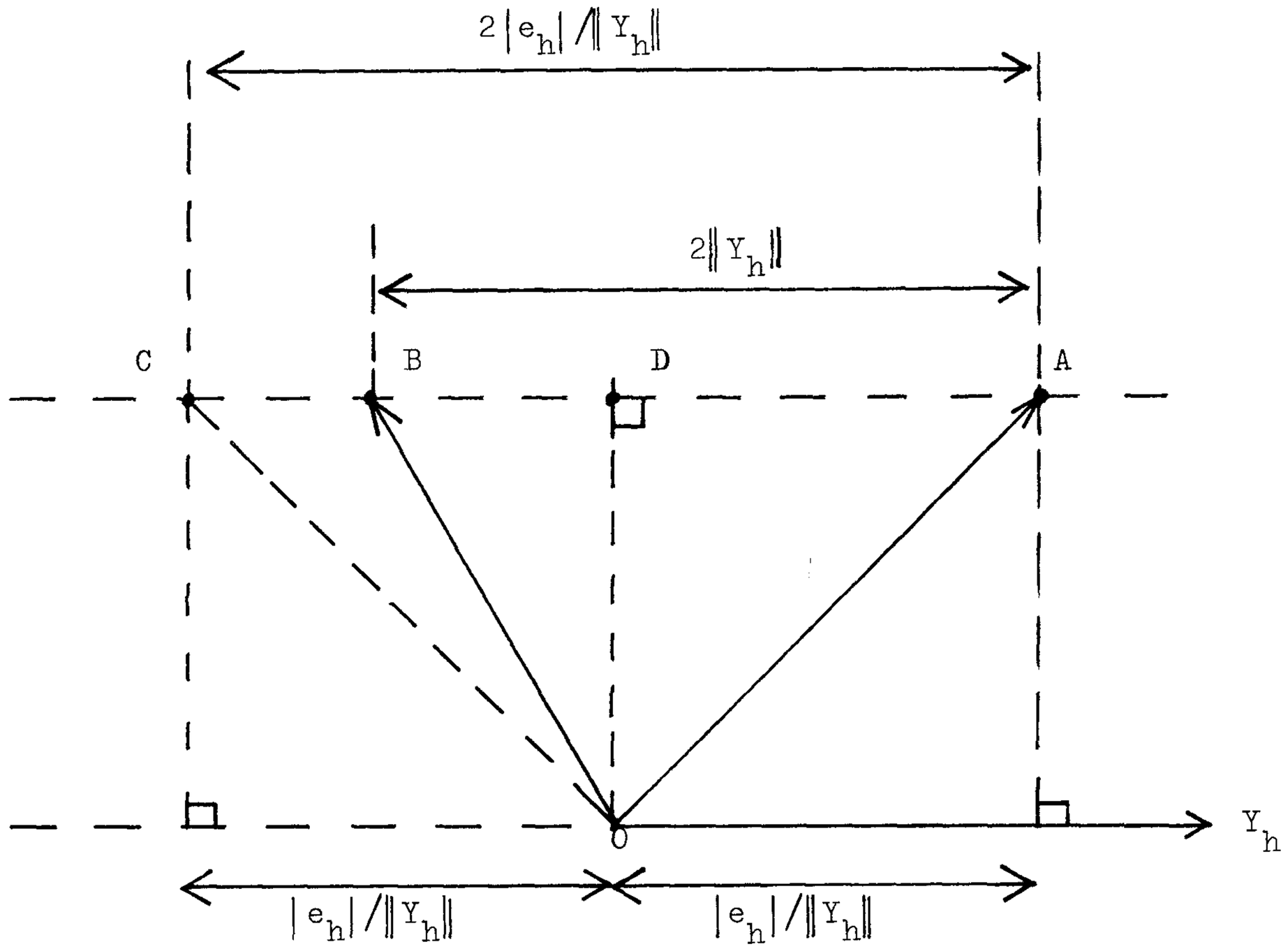


Fig. A1.1 R_a and Y_h (eqn. A1.4) represented as vectors in the n-dimensional Euclidean vector space.

OA : vector of R_a before it is updated

OB : updated vector of R_a

APPENDIX A2

PROOF OF SYSTEMS 3.1 - 3.4 (CHAPTER 3) BEING ABLE TO ACHIEVE ERROR-FREE DETECTION IN THE ABSENCE OF NOISE

This appendix shows that, in the detection of s_1 (eqn. 3.5) from R' (eqns. 3.7 and 3.8), each of the systems 3.1 - 3.4 (chapter 3) is able to achieve, in the absence of noise,

$$X = S \qquad \text{(A2.1)}$$

and hence an error-free detection, bearing in mind that the detected value of the first component s_1 of the n-component vector S (eqn. 3.5) is taken to have the value of the first component x_1 of the n-component vector X (eqn. 3.13) obtained at the end of the detection process here.

Assuming that all previous data symbols $\{s_i\}$ have been detected correctly so that the vector R' is now given by eqn. 3.8. In the absence of noise, eqn. 3.8 reduces to

$$\begin{aligned} R' &= SY \\ &= s_1Y_1 + s_2Y_2 + \cdots + s_nY_n \end{aligned} \qquad \text{(A2.2)}$$

where $Y_1, Y_2, \cdots, Y_n$ are the n n-component vectors defined by eqn. 3.9. The vector R_a (eqn. 3.22) now becomes

$$\begin{aligned} R_a &= R' - XY \\ &= SY - XY \\ &= (s_1 - x_1)Y_1 + (s_2 - x_2)Y_2 + \cdots + (s_n - x_n)Y_n \end{aligned} \qquad \text{(A2.3)}$$

In any of the systems 3.1 - 3.4, the initial vector X_o of X is set to

$$X_o = \begin{bmatrix} s'_2 & s'_3 & \cdots & s'_n & 0 \end{bmatrix} \qquad (A2.4)$$

where $s'_2, s'_3, \cdots, s'_n$ are the detected values of the corresponding data symbols $s_2, s_3, \cdots, s_n$ obtained at the end of the previous detection process, bearing in mind that these data symbols are renamed as $s_1, s_2, \cdots, s_{n-1}$ respectively for the next detection process. Assuming that the values of $s'_2, s'_3, \cdots, s'_n$ in eqn. A2.4 have been determined correctly, the initial vector of R_a (eqn. A2.3) now becomes

$$\begin{aligned} R_a &= SY - X_o Y \\ &= s_n Y_n \end{aligned} \qquad (A2.5)$$

Tne iterative process of system 3.1 now begins with the evaluation of the quantity e_1 (eqn. 3.24) as $R_a Y_1^T$, so that from eqn. A2.5,

$$\begin{aligned} e_1 &= R_a Y_1^T \\ &= s_n (Y_n Y_1^T) \end{aligned} \qquad (A2.6)$$

which, in turn, is used to update the value of x_1 (Table 3.2 and eqn. 3.30). It can be seen from Table 3.2 and eqn. 3.30 that, when $x_h = \pm 1$, the value of x_h remains unchanged during its updating process so long as

$$|e_h| \leq \|Y_h\|^2 \qquad (A2.7)$$

for $h = 1, 2, \cdots, n$. The value of $|e_1|$ in eqn. A2.6 is given by

$$|e_1| = |s_n Y_n Y_1^T| = |Y_n Y_1^T| \qquad (A2.8)$$

since s_n can only have a value of 1 or -1. Furthermore, it can be seen

from eqn. 3.9 that, $\|Y_n\| > \|Y_h\|$ so that

$$|Y_n Y_h^T| > |Y_h Y_h^T| = \|Y_h\|^2 \qquad \text{(A2.9)}$$

for $h = 1, 2, \cdots, n-1$. It therefore follows from eqns. A2.7 - A2.9 that, $|e_1| < \|Y_1\|^2$ here, so that the value of x_1 (being equal to s_2') and hence the vector R_a (being updated by using eqn. 3.31) are unchanged in the updating process here. The iterative process next evaluates the quantity e_2 as $R_a Y_2^T$, and from eqn. A2.5,

$$\begin{aligned} e_2 &= R_a Y_2^T \\ &= s_n(Y_n Y_2^T) \end{aligned} \qquad \text{(A2.10)}$$

for the updating process for x_2 (Table 3.2, eqns. 3.30 and 3.31). Again, it can be seen from eqns. A2.7, A2.9, and A2.10 that, $|e_2| < \|Y_2\|^2$ here, so that the value of x_2 (being equal to s_3') and hence the vector R_a are unchanged in the updating process here. Following the same argument just described, it can be shown that the values of x_3 $(= s_4')$, x_4 $(= s_5')$, $\cdots$, x_{n-1} $(= s_n')$ and the vector R_a (being given by eqn. A2.5) are all unchanged following the evaluations of $e_3, e_4, \cdots, e_{n-1}$. The quantity e_n is now evaluated as $R_a Y_n^T$, and from eqn. A2.5,

$$\begin{aligned} e_n &= R_a Y_n^T \\ &= s_n(Y_n Y_n^T) \\ &= s_n\|Y_n\|^2 \end{aligned} \qquad \text{(A2.11)}$$

Since the initial value of x_n here is zero (eqn. A2.4), the updated value of x_n is determined as 1 if $e_n > 0$, and as -1 if $e_n \leq 0$ (Table 3.2).

Furthermore, since the value of e_n is given by eqn. A2.11, the updated value of x_n here must be equal to the value of s_n. Consequently, all the values of $x_1, x_2, \cdots, x_n$ are now equal to the corresponding values of $s_1, s_2, \cdots, s_n$ (that is, X = S), and the vector R_a (eqn. A2.3) is now equal to zero. In the subsequent iterative cycles of the iterative process of system 3.1 here, the values of $x_1, x_2, \cdots, x_n$ are always unchanged since the inequality of (A2.7) is always satisfied here, bearing in mind that e_h $(1 \leq h \leq n)$, being equal to $R_a Y_h^T$, is always equal to zero here. Thus, X = S here so long as all the previous data symbols have been detected correctly. Since a known sequence of more than g data symbols is transmitted at the start of the detection process of system 3.1, therefore X = S here and the following data symbols now continue to be detected correctly and so on. Error-free detection in the absence of noise is therefore always achieved by system 3.1.

It can be seen from above that, the value of e_h $(1 \leq h \leq n)$ is always equal to $s_n Y_n Y_h^T$ before the value of x_n is determined, so that the value of x_n is always determined correctly as the value of s_n. Furthermore, the value of e_h $(1 \leq h \leq n)$ is always equal to zero after the value of x_n has been determined. In any case, the inequality of (A2.7) is always satisfied whenever $x_h = \pm 1$, so that the value of x_h once being determined as 1 or -1 will not be changed anymore during the iterative process, regardless of whether the iterative process operates from x_1 to x_n or from x_n to x_1. The argument described above for system 3.1 can therefore be used to show that each of the systems 3.2 - 3.4 is also able to achieve X = S and hence error-free detection in the absence of noise, bearing in mind that systems 3.1 - 3.4 differ only in the 'direction' of operation (that is, operating from x_1 to x_n or from x_n to x_1).

APPENDIX A3

DERIVATION OF THE CONDITION FOR SYSTEM 3.5 (CHAPTER 3) TO ACHIEVE ERROR-FREE DETECTION IN THE ABSENCE OF NOISE

This appendix shows that, in the detection of s_1 (eqn. 3.5) from R' (eqns. 3.7 and 3.8), the condition for system 3.5 (chapter 3) to achieve error-free detection (or X = S) in the absence of noise is

$$\| Y_h \|^2 \quad > \quad \sum_{j=1}^{n-h} | Y_{h+j} Y_h^T | \qquad \text{(A3.1)}$$

for $h = n-f+1, n-f+2, \cdots, n-1$ where $\| Y_h \|$ is the Euclidean norm or length of the n-component vector Y_h (eqn. 3.9), $| Y_{h+j} Y_h^T |$ is the absolute value of the scalar quantity $Y_{h+j} Y_h^T$, and f is an integer between 1 to n.

In the absence of noise, and assuming that all previous data symbols $\{s_i\}$ have been detected correctly, the n-component vector R' is given by (eqn. A2.2)

$$R' = s_1 Y_1 + s_2 Y_2 + \cdots + s_n Y_n \qquad \text{(A3.2)}$$

and the n-component vector R_a is given by (eqn. A2.3)

$$R_a = (s_1 - x_1) Y_1 + (s_2 - x_2) Y_2 + \cdots + (s_n - x_n) Y_n \qquad \text{(A3.3)}$$

where $s_1, s_2, \cdots, s_n$ are the n components of the data-symbol vector S (eqn. 3.5), and $x_1, x_2, \cdots, x_n$ are the n components of the vector X (eqn. 3.13). In system 3.5, the initial vector X_o of X is set to

$$X_o = \left[s'_2 \quad s'_3 \quad \cdots \quad s'_{n-f+1} \quad \overbrace{0 \quad \cdots \quad 0}^{f} \right] \qquad \text{(A3.4)}$$

where $s'_2, s'_3, \cdots, s'_{n-f+1}$ are the detected values of the corresponding

data symbols $s_2, s_3, \cdot\cdot, s_{n-f+1}$ obtained at the end of the previous detection process, bearing in mind that these data symbols are shifted one place forward in S and renamed as $s_1, s_2, \cdot\cdot, s_{n-f}$ respectively for the next detection process. Assuming that the values of $s'_2, s'_3, \cdot\cdot s'_{n-f+1}$ in eqn. A3.4 have been determined correctly, the initial vector of R_a (eqn. A3.3) now becomes

$$\begin{aligned} R_a &= R' - X_o Y \\ &= s_{n-f+1} Y_{n-f+1} + s_{n-f+2} Y_{n-f+2} + \cdot\cdot + s_n Y_n \end{aligned} \qquad (A3.5)$$

The iterative process of system 3.5 now begins with the evaluation of the quantity e_{n-f+1} (eqn. 3.24) as $R_a Y^T_{n-f+1}$, and from eqn. A3.5,

$$e_{n-f+1} = s_{n-f+1} \| Y_{n-f+1} \|^2 + \sum_{h=n-f+2}^{n} (s_h Y_h) Y^T_{n-f+1} \qquad (A3.6)$$

The value of x_{n-f+1} is then determined as 1 if $e_{n-f+1} > 0$, and as -1 if $e_{n-f+1} \leqslant 0$. It can be seen from eqn. A3.6 that, the value of x_{n-f+1} is always determined correctly as the value of s_{n-f+1} here so long as

$$\| Y_{n-f+1} \|^2 > \sum_{h=n-f+2}^{n} | Y_h Y^T_{n-f+1} | \qquad (A3.7)$$

Having updated the value of x_{n-f+1} and the vector R_a (eqn. 3.31), the iterative process next evaluates the quantity e_{n-f+2} as $R_a Y^T_{n-f+2}$. Clearly, if the inequality of (A3.7) is satisfied, then $x_{n-f+1} = s_{n-f+1}$, and the component $s_{n-f+1} Y_{n-f+1}$ is removed from R_a, so that the value of e_{n-f+2} now reduces to

$$e_{n-f+2} = s_{n-f+2} \| Y_{n-f+2} \|^2 + \sum_{h=n-f+3}^{n} (s_h Y_h) Y^T_{n-f+2} \qquad (A3.8)$$

The value of x_{n-f+2} is then determined as 1 if $e_{n-f+2} > 0$, and as -1 if $e_{n-f+2} \leq 0$. The situation here obviously resembles that for the updating process for x_{n-f+1}, and the value of x_{n-f+2} is thus always determined correctly as the value of s_{n-f+2} here so long as

$$\| Y_{n-f+2} \|^2 > \sum_{h=n-f+3}^{n} | Y_h Y_{n-f+2}^T | \qquad \text{(A3.9)}$$

which resembles the inequality of (A3.7). Following the same argument just described, it can be seen that the values of x_{n-f+1}, x_{n-f+2}, $\cdots$, x_{n-1} are always determined correctly as the corresponding values of s_{n-f+1}, s_{n-f+2}, $\cdots$, s_{n-1} in the first iterative cycle of the iterative process here so long as

$$\| Y_h \|^2 > \sum_{j=1}^{n-h} | Y_{h+j} Y_h^T | \qquad \text{(A3.10)}$$

for $h = n-f+1, n-f+2, \cdots, n-1$. Thus, if this inequality (A3.10) is satisfied, then the vector X becomes

$$X = \begin{bmatrix} s_1 & s_2 & \cdots & s_{n-1} & 0 \end{bmatrix} \qquad \text{(A3.11)}$$

following the updating process for x_{n-1} in the first iterative cycle here. This vector of X is identical to the initial vector of X used in system 3.1 (in the absence of noise). Since system 3.5 now operates in the same way as system 3.1 in the subsequent operations, the same argument described in appendix A2 for system 3.1 can be applied here to show that, system 3.5 is now always able to achieve X = S and hence error-free detection. Consequently, system 3.5 is always able to achieve X = S and hence error-free detection in the absence of noise so long as the condition of (A3.10) (being identical to (A3.1)) is satisfied.

APPENDIX A4

DERIVATION OF eqn. 3.70 (CHAPTER 3)

This appendix shows that

$$Y_j Z_h^T = 0 \qquad \text{if} \quad j < h \tag{A4.1}$$

$$> 0 \qquad \text{if} \quad j = h \tag{A4.2}$$

for $h = 1, 2, \cdots, n$ where $\{Y_j\}$ are the $(1 \times n)$ row matrices defined by eqn. 3.9, and $\{Z_h^T\}$ are the transposes of the corresponding $(1 \times n)$ row matrices $\{Z_h\}$ defined by eqns. 3.65 and 3.66.

It can be seen from eqn. 3.65 that

$$Y_j = Z_j' + \sum_{i=1}^{j-1} \frac{1}{\|Z_i'\|^2} Y_j (Z_i')^T Z_i' \tag{A4.3}$$

and

$$(Z_h')^T = \left[Y_h - \sum_{k=1}^{h-1} \frac{1}{\|Z_k'\|^2} Y_h (Z_k')^T Z_k' \right]^T$$

$$= Y_h^T - \sum_{k=1}^{h-1} \frac{1}{\|Z_k'\|^2} (Z_k')^T (Z_k') Y_h^T \tag{A4.4}$$

for any possible values of j and h. Thus,

$$Y_j (Z_h')^T = Y_j Y_h^T - \sum_{k=1}^{h-1} \frac{1}{\|Z_k'\|^2} Y_j (Z_k')^T (Z_k') Y_h^T \tag{A4.5}$$

or

$$Y_j (Z_h')^T = (Z_j') Y_h^T + \sum_{i=1}^{j-1} \frac{1}{\|Z_i'\|^2} Y_j (Z_i')^T (Z_i') Y_h^T$$

$$- \sum_{k=1}^{h-1} \frac{1}{\|Z_k'\|^2} Y_j (Z_k')^T (Z_k') Y_h^T \tag{A4.6}$$

Consider first the case when $j < h$. Eqn. A4.6 may be reduced to

$$Y_j(Z'_h)^T = (Z'_j)Y_h^T - \sum_{k=j}^{h-1} \frac{1}{\|Z'_k\|^2} Y_j(Z'_k)^T(Z'_k)Y_h^T$$

$$= (Z'_j)Y_h^T - \sum_{k=j}^{h-1} \frac{1}{\|Z'_k\|^2} (Z'_j) + \sum_{i=1}^{j-1} \frac{1}{\|Z'_i\|^2} Y_j(Z'_i)^T(Z'_i) \Big](Z'_k)^T(Z'_k)Y_h^T$$

(A4.7)

Since $(Z'_i)(Z'_k)^T = 0$ for $i \neq k$ (eqn. 3.68), eqn. A4.7 may be further reduced to

$$Y_j(Z'_h)^T = (Z'_j)Y_h^T - \frac{1}{\|Z'_j\|^2} (Z'_j)(Z'_j)^T(Z'_j)Y_h^T$$

$$= (Z'_j)Y_h^T - \frac{1}{\|Z'_j\|^2} \|Z'_j\|^2 (Z'_j)Y_h^T$$

$$= 0 \qquad \text{(A4.8)}$$

It can now be seen from eqns. 3.66 and A4.8 that

$$Y_j Z_h^T = Y_j(Z'_h)^T \frac{1}{\|Z'_h\|} = 0 \qquad \text{(A4.9)}$$

which is identical to eqn. A4.1.

Consider now the case when $j = h$. Eqn. A4.5 may be reduced to

$$Y_h(Z'_h)^T = \|Y_h\|^2 - \sum_{k=1}^{h-1} \frac{1}{\|Z'_k\|^2} Y_h(Z'_k)^T(Z'_k)Y_h^T$$

$$= \|Y_h\|^2 - \sum_{k=1}^{h-1} \Big[Y_h(Z_k)^T\Big]\Big[(Z_k)Y_h^T\Big]$$

$$= \|Y_h\|^2 - \sum_{k=1}^{h-1} \|Y_h Z_k^T\|^2 \qquad \text{(A4.10)}$$

If Y_h and Z_k are each treated as a n-component vector in the n-dimensional Euclidean vector space, then $Y_h Z_k^T$ is the orthogonal projection of the vector Y_h onto the unit vector Z_k, as is shown in Fig. A4.1. Moreover, since the n unit vectors $Z_1, Z_2, \cdots, Z_n$ are orthogonal to each other (eqns. 3.67 and 3.69), they necessarily form a complete set of basis in the n-dimensional Euclidean vector space here.(E5) This means that, the Euclidean norm or length of the vector Y_h (that is, $\|Y_h\|$) is such that

$$\|Y_h\|^2 = \sum_{k=1}^{n} |Y_h Z_k^T|^2 \qquad \text{(A4.11)}$$

Fig. A4.2 illustrates the meaning of eqn. A4.11 for n = 2. It can now be seen from eqns. A4.10 and A4.11 that

$$|Y_h (Z_h')^T| > 0 \qquad \text{(A4.12)}$$

since $h \leq n$. Eqns. 3.66 and A4.12 can now be combined to give

$$Y_h Z_h^T \|Z_h\| > 0$$

or

$$Y_h Z_h^T > 0 \qquad \text{(A4.13)}$$

which is identical to eqn. A4.2.

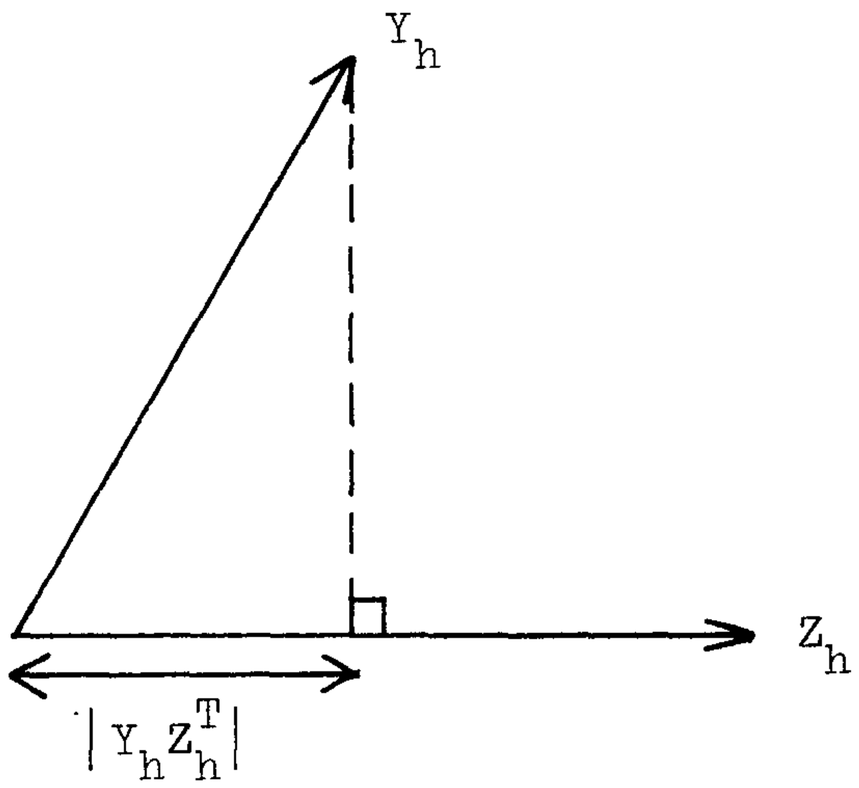


Fig. A4.1 Orthogonal projection of the vector Y_h onto the unit vector Z_k in the Euclidean vector space.

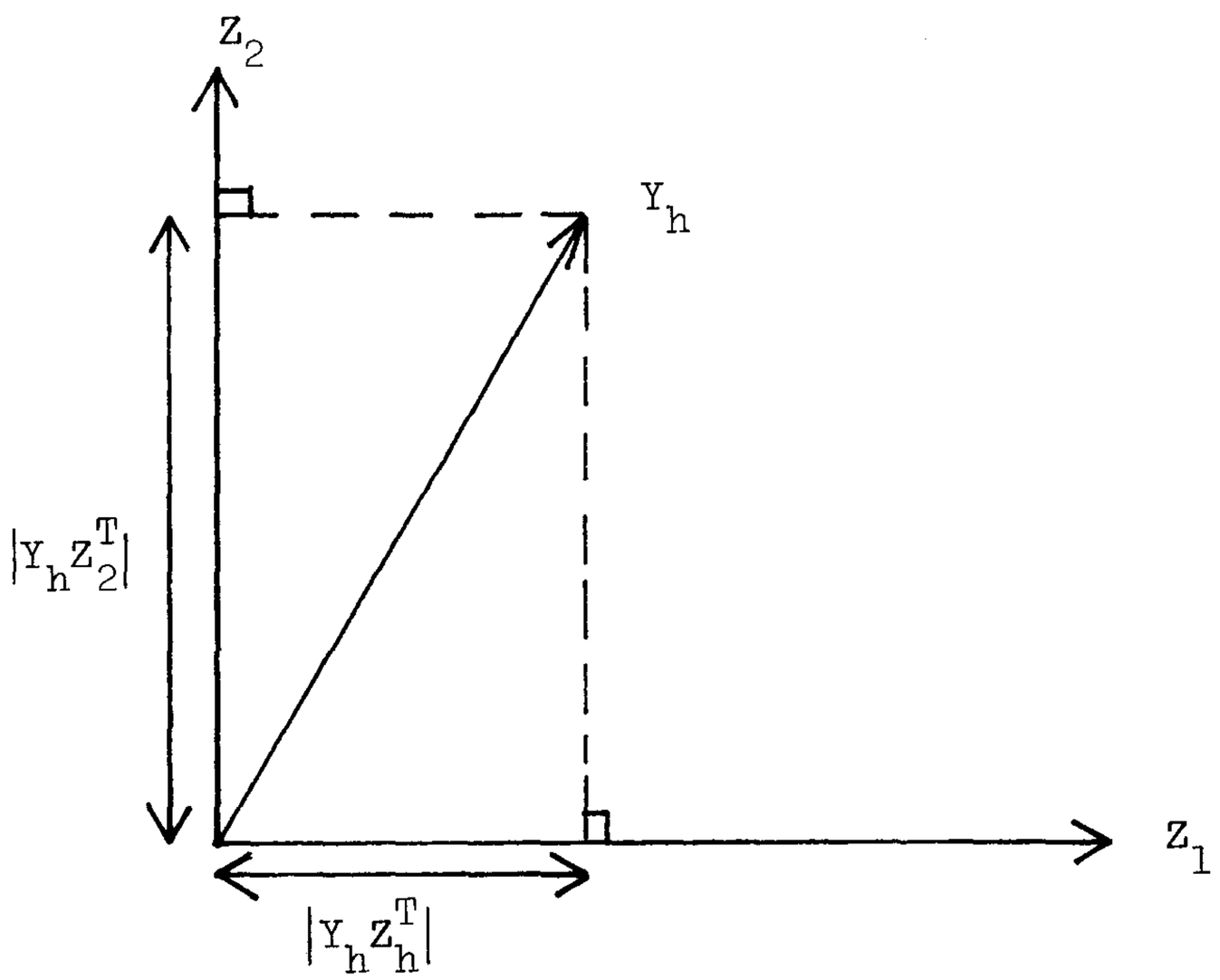


Fig. A4.2 A 2-dimensional Euclidean vector space containning the vector Y_h, and the unit vectors Z_1 and Z_2.

APPENDIX A5

DIVERGENCY OF THE ITERATIVE PROCESSES OF SYSTEMS 8 - 11 (CHAPTER 3)

This appendix shows that, divergence can occur in the iterative process of any of the systems 8 - 11 (chapter 3). The iterative process here employs the arrangement of Fig. 3.14, and divergence is defined to have occurred when the value of $\|R_a\|$ increases during the iterative process, where R_a is the n-component vector defined by eqn. 3.22.

In the iterative process here, the values of the n components $x_1, x_2, \cdots, x_n$ of the vector X (eqn. 3.13) are updated sequentially using Tables 3.1, 3.2, 3.8, and eqn. 3.91, and the vector R_a is updated using eqn. 3.31. A constraint is placed on the values of the components $\{x_h\}$ $(1 \leq h \leq n)$, so that each component here can only have a value of -1, 0, or 1 during the iterative process.

Consider first the case when $x_h = \pm 1$, for $1 \leq h \leq n$. In the updating process for x_h, the quantity e_h (eqn. 3.72) is first evaluated at the output of the network Z_h^T shown in Fig. 3.14 as

$$e_h = R_a Z_h^T \qquad (A5.1)$$

where Z_h^T is the transpose of Z_h. The increment Δx_h of x_h is next determined using Table 3.2. Thus, if x_h has the same sign as that of e_h, or if $|e_h| \leq |Y_h Z_h^T|$ $(Y_h \neq Z_h)$, then Δx_h is set to zero and the vector R_a is unchanged so that neither divergence nor convergence can occur here. Y_h is, of course, the n-component vector defined by eqn. 3.9. If, however, x_h has a different sign from that of e_h and if $|e_h| > |Y_h Z_h^T|$, then Δx_h is set to 2 or -2 depending on whether x_h is negative or positive.

The vector R_a is now updated as (eqn. 3.31)

$$(R_a)_{new} = R_a \pm 2Y_h \qquad (A5.2)$$

In this case, divergence can sometimes occur as is shown below. Fig. A5.1 shows the n-component vectors Y_h and Z_h in the n-dimensional Euclidean vector space. In Fig. A5.1, OP represents the vector of R_a before it is updated, and OQ represents the updated vector of R_a. The value of $\|R_a\|$ is here given by the corresponding length of OP or OQ. The straight line PQ is parallel to the direction of the vector Y_h as is implied by eqn. A5.2. The point P here is restricted to lie on the right hand side of CD, since it is considered here that $|e_h| > |Y_h Z_h^T|$. It can be seen from Fig. A5.1 that, if P lies between G and H, then OQ has a larger length or $\|R_a\|$ as compared to that of OP, which necessarily means that divergence will occur here. In general, it can be seen that, divergence occurs in the iterative process here ($x_h = \pm 1$) so long as P lies in the shaded area shown in Fig. A5.1.

Consider now the case when $x_h = 0$. Table 3.1 shows that if $|e_h| \leq t_h$, then Δx_h is set to zero and the vector R_a remains unchanged so that neither divergence nor convergence can occur here. The value of t_h is here equal to 0, or $\frac{1}{4}|Y_h Z_h^T|$, or $\frac{1}{2}|Y_h Z_h^T|$, or $|Y_h Z_h^T|$ as is shown in Table 3.8. If, however, $|e_h| > t_h$, then Δx_h is set to 1 or -1 depending on whether e_h is positive or negative, and the vector R_a is updated as (eqn. 3.31)

$$(R_a)_{new} = R_a \pm Y_h \qquad (A5.3)$$

which resembles eqn. A5.2. The same argement used for the case when $x_h = \pm 1$ can thus be applied here to show that divergence can occur for the case when $x_h = 0$.

Although the argument used in this appendix assumes that all the vectors R_a, Y_h, and Z_h lie in the same 2-dimensional plane shown in Fig . A5.1 , it can be seen that the argument also applies for the general case when these vectors are not all containned in the same 2-dimensional plane, so that divergence can occur in the iterative process here for any given vectors of R_a, Y_h, and Z_h so long as $Y_h \neq Z_h$.

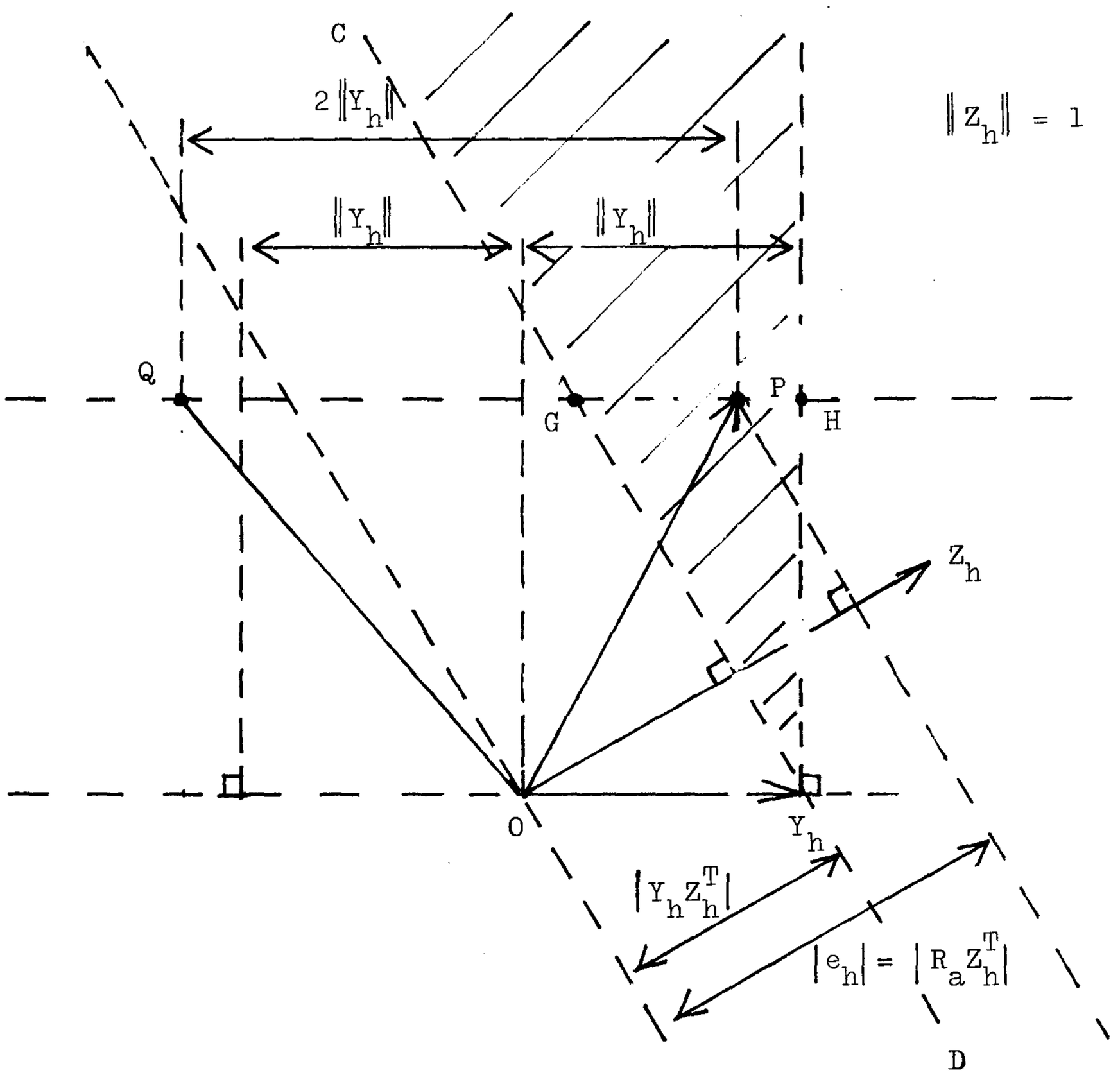


Fig. A5.1 R_a, Y_h, and Z_h (eqn. A5.2) represented as vectors in the n-dimensional Euclidean vector space.

OP : vector of R_a before it is updated

OQ : updated vector of R_a

APPENDIX A6

PROOF OF SYSTEM 8 (CHAPTER 3) BEING ABLE TO ACHIEVE ERROR-FREE DETECTION IN THE ABSENCE OF NOISE

This appendix shows that, in the detection of s_1 (eqn. 3.5) from R' (eqns. 3.7 and 3.8), system 8 (chapter 3) is able to achieve error-free detection in the absence of noise.

In the absence of noise, and assuming that all previous data symbols $\{s_i\}$ have been detected correctly, the vector R' now becomes

$$\begin{aligned} R' &= SY \\ &= s_1Y_1 + s_2Y_2 + \cdots + s_nY_n \end{aligned} \qquad \text{(A6.1)}$$

where $s_1, s_2, \cdots, s_n$ are the n components of the data-symbol vector S (eqn. 3.5), and $\{Y_j\}$ are the n-component vectors defined by eqn. 3.9. The quantity e_h (eqn. 3.72) in system 8 now becomes

$$\begin{aligned} e_h &= (R' - XY)Z_h^T \\ &= (SY - XY)Z_h^T \\ &= (s_1 - x_1)Y_1Z_h^T + (s_2 - x_2)Y_2Z_h^T + \cdots + (s_n - x_n)Y_nZ_h^T \end{aligned} \qquad \text{(A6.2)}$$

for $h = 1, 2, \cdots, n$ where $x_1, x_2, \cdots, x_n$ are the n components of the vector X (eqn. 3.13), and $\{Z_h^T\}$ are the transposes of the corresponding $\{Z_h\}$ defined by eqns. 3.65 and 3.66. Eqn. 3.70 suggests that, eqn. A6.2 may be reduced to

$$e_h = (s_h - x_h)\left|Y_hZ_h^T\right| + (s_{h+1} - x_{h+1})Y_{h+1}Z_h^T + \cdots + (s_n - x_n)Y_nZ_h^T \qquad \text{(A6.3)}$$

for $h = 1, 2, \cdots, n$.

In the iterative process of system 8.1 with any of the versions a, b, c, and d given in Table 3.8, the values of $x_1, x_2, \cdots, x_n$ are updated according to Table 3.1 in the first iterative cycle, and according to Table 3.2 in each of the subsequent iterative cycles. The values of $x_1, x_2, \cdots, x_n$ are here initially set to zero. Thus, in the first iterative cycle here, the value of e_h is, from eqn. A6.3, given by

$$e_h = s_h|Y_h Z_h^T| + s_{h+1}Y_{h+1}Z_h^T + \cdots + s_n Y_n Z_h^T \qquad (A6.4)$$

for $h = 1, 2, \cdots, n$. Clearly, $e_n = s_n|Y_n Z_n^T|$ here, so that the value of x_n is always determined correctly as the value of s_n in the first iterative cycle here as can be seen from Tables 3.1 and 3.8. This means that, at the end of the first iterative cycle, the vector X in system 8.1 becomes

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-1} & s_n \end{bmatrix} \qquad (A6.5)$$

where $x_1, x_2, \cdots, x_{n-1}$ may have any of the values 1, 0, and -1. In the second iterative cycle, the value of e_h is now given by

$$e_h = (s_h - x_h)|Y_h Z_h^T| + (s_{h+1} - x_{h+1})Y_{h+1}Z_h^T + \cdots + (s_{n-1} - x_{n-1})Y_{n-1}Z_h^T \qquad (A6.6)$$

for $h = 1, 2, \cdots, n-1$, and is equal to zero for $h = n$. The value of e_{n-1} is now given by $(s_{n-1} - x_{n-1})|Y_{n-1}Z_{n-1}^T|$. Thus, if $x_{n-1} = s_{n-1}$, then $e_{n-1} = 0$ and from Table 3.2, the value of x_{n-1} must now remain unchanged during the updating process for x_{n-1}. If $x_{n-1} = -s_{n-1} = -1$, then $e_{n-1} = 2|Y_{n-1}Z_{n-1}^T|$ so that, from Table 3.2, the value of x_{n-1} must now be set to 1 which is the value of s_{n-1} here. Similarly, if

$x_{n-1} = -s_{n-1} = 1$, then $e_{n-1} = -2|Y_{n-1}Z_{n-1}^T|$ so that, from Table 3.2, the value of x_{n-1} must now be set to -1 which is the value of s_{n-1} here. Finally, if $x_{n-1} = 0$, then $e_{n-1} = s_{n-1}|Y_{n-1}Z_{n-1}^T|$, and from Table 3.2, the value of x_{n-1} is set to the value of s_{n-1}. It therefore follows that, in the second iterative cycle here, the value of x_{n-1} is always updated to be equal to the value of s_{n-1}. The value of x_n, being set to the value of s_n in the first iterative cycle, remains to be unchanged in the second and subsequent iterative cycles since the value of e_n is now always equal to zero (see Table 3.2). The vector X obtained at the end of the second iterative cycle here is therefore given by

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-2} & s_{n-1} & s_n \end{bmatrix} \qquad (A6.7)$$

where $x_1, x_2, \cdots, x_{n-2}$ may have any of the two values 1 and -1 which are the two possible values of the data symbols $\{s_h\}$. Each subsequent iterative cycle here operates in exactly the same way as for the second iterative cycle. Consequently, it can be seen from the argument given above that, at the end of the nth iterative cycle here, the vector X becomes

$$X = \begin{bmatrix} s_1 & s_2 & \cdots & s_n \end{bmatrix} \qquad (A6.8)$$

which is obviously identical to the data-symbol vector S (eqn. 3.5). All the values of $e_1, e_2, \cdots, e_n$ are now equal to zero, and thus, from Table 3.2, the vector X (= S here) must always remain to be unchanged in the subsequent iterative cycles. Thus, if n_c (number of iterative cycle) is equal to or larger than n, then X = S so long as all the previous data symbols $\{s_i\}$ have been detected correctly. Since a known sequence of more than g data symbols is transmitted at the start of

the detection process of system 8.1, therefore X = S here (assuming that $n_c \geq n$) and the following data symbols now continue to be detected correctly and so on. Error-free detection in the absence of noise is therefore always achieved by system 8.1 so long as $n_c \geq n$.

Since system 8.2 differs from system 8.1 only in the 'direction' of operation of the iterative process (that is, system 8.1 operates from x_1 to x_n whereas system 8.2 operates from x_n to x_1 for each iterative cycle), therefore the same argument used for system 8.1 can be used to show that error-free detection in the absence of noise is always achieved by system 8.2 so long as $n_c \geq 1$.

APPENDIX A7

DERIVATION OF THE CONDITIONS FOR SYSTEMS 9 - 11 (CHAPTER 3) TO ACHIEVE ERROR-FREE DETECTION IN THE ABSENCE OF NOISE

This appendix shows that, in the detection of s_1 (eqn. 3.5) from R' (eqns. 3.7 and 3.8), the condition for error-free detection to be achieved in the absence of noise is

$$|Y_1 Z_1^T| > 2 \sum_{j=2}^{n} |Y_j Z_1^T| \qquad (A7.1)$$

for any of the systems 9 and 11, and is

$$|Y_h Z_h^T| > 2 \sum_{j=h+1}^{n} |Y_j Z_h^T| \qquad (A7.2)$$

for $h = 2, 3, \cdots, n$, for system 10, where $\{Y_j\}$ are the n-component vectors defined by eqn. 3.9 and $\{Z_h^T\}$ are the transposes of the corresponding $\{Z_h\}$ derived in section 3.13 (chapter 3).

In the absence of noise, and assuming that all previous data symbols $\{s_i\}$ have been detected correctly, the value of e_h in eqn. 3.79 becomes

$$e_h = (s_h - x_h)\, Y_h Z_h^T + \sum_{\substack{j=1 \\ j \neq h}}^{n} (s_j - x_j) Y_j Z_h^T \qquad (A7.3)$$

for $h = 1, 2, \cdots, n$ where $|Y_h Z_h^T|$ is the absolute value of $Y_h Z_h^T$, and the $\{x_j\}$ may have any of the values 1, 0, and -1, bearing in mind that 1 and -1 are two possible values of $\{s_j\}$ here. Assuming for the moment that the values of $x_1, x_2, \cdots, x_{h-1}$ have been determined

correctly (that is, $x_j = s_j$ for $h = 1, 2, \cdots, h-1$), eqn. A7.3 now reduces to

$$e_h = (s_h - x_h)\left|Y_h Z_h^T\right| + \sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \qquad (A7.4)$$

for $h = 1, 2, \cdots, n$. Since the value of $(s_j - x_j)$ can only be one of the values ± 2, ± 1, and 0, the maximum magnitude of the intersymbol interference component of e_h in eqn. A7.4 is equal to $2\sum_{j=h+1}^{n}\left|Y_j Z_h^T\right|$. That is,

$$2\sum_{j=h+1}^{n}\left|Y_j Z_h^T\right| \geq \left|\sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T\right| \qquad (A7.5)$$

for $h = 1, 2, \cdots, n$. In the iterative process (Fig. 3.14) of any of the systems 9 - 11, the value of x_h $(1 \leq h \leq n)$ is determined from the value of e_h using Tables 3.1, 3.2, 3.8, and eqn. 3.91. However, only those updating processes for $\{x_h\}$ that involve the use of Table 3.2 and eqn. 3.91 are considered here. This means that, each of the systems 9 - 11 with any of the versions b, c, and d (Table 3.8) is assumed to operate with at least two iterative cycles here. It can be seen from Table 3.2 that, the value of x_h before it is updated here, can be any of the values 1, 0, and -1. Consider first the case when $x_h = 0$. The value of e_h in eqn. A7.4 now reduces to

$$e_h = s_h\left|Y_h Z_h^T\right| + \sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \qquad (A7.6)$$

for $h = 1, 2, \cdots, n$. The updated value of x_h here is set to 1 if $e_h > 0$, and to -1 if $e_h \leq 0$. It can be seen from eqn. A7.6 that, when $s_h = 1$, $e_h > 0$ if

$$\left|Y_h Z_h^T\right| > -\sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \qquad (A7.7)$$

and when $s_h = -1$, $e_h \leq 0$ if

$$\left|Y_h Z_h^T\right| \geq \sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \qquad (A7.8)$$

Furthermore, it can be seen from eqn. A7.5 that, the inequalities of (A7.7) and (A7.8) are always satisfied so long as

$$\left|Y_h Z_h^T\right| > 2\sum_{j=h+1}^{n}\left|Y_j Z_h^T\right| \qquad (A7.9)$$

which is therefore the condition for the updated value of x_h to be equal to the value of s_h here. Consider next the case when $x_h = \pm 1$ before it is updated. The value of e_h in eqn. A7.4 now reduces to

$$\begin{aligned} e_h &= 2\left|Y_h Z_h^T\right| + \sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \quad , \quad \text{if} \quad x_h = -s_h = -1 \\ &= -2\left|Y_h Z_h^T\right| + \sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \quad , \quad \text{if} \quad x_h = -s_h = 1 \\ &= 0 + \sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \quad , \quad \text{if} \quad x_h = s_h = \pm 1 \end{aligned} \qquad (A7.10)$$

for $h = 1, 2, \cdots, n$. The updating process for x_h here is such that, when $x_h = 1$, the value of x_h remains unchanged unless $e_h < -\left|Y_h Z_h^T\right|$ when the updated value of x_h is set to -1, and when $x_h = -1$, the value of x_h remains unchanged unless $e_h > \left|Y_h Z_h^T\right|$ when the updated value of x_h is set to 1. It can now be seen from eqn. A7.10 that, when $x_h = -s_h = -1$, the updated value of x_h is equal to s_h if $e_h > \left|Y_h Z_h^T\right|$ or

$$\left|Y_h Z_h^T\right| > -\sum_{j=h+1}^{n}(s_j - x_j)Y_j Z_h^T \qquad (A7.11)$$

and when $x_h = -s_h = 1$, the updated value of x_h is equal to s_h if $e_h < -|Y_h Z_h^T|$ or

$$|Y_h Z_h^T| > \sum_{j=h+1}^{n} (s_j - x_j) Y_j Z_h^T \qquad \text{(A7.12)}$$

and finally when $x_h = s_h$, the updated value of x_h remains unchanged if $|e_h| \leq |Y_h Z_h^T|$ or

$$|Y_h Z_h^T| \geq \sum_{j=h+1}^{n} (s_j - x_j) Y_j Z_h^T \qquad \text{(A7.13)}$$

It can be seen from eqn. A7.5 that, the inequalities of (A7.11) - (A7.13) are always satisfied so long as the inequality of (A7.9) is satisfied, and the inequality of (A7.9) is therefore the condition for the updated value of x_h to be equal to the value of s_h here. Consequently, so long as the values of $x_1, x_2, \cdots, x_{h-1}$ are equal to the corresponding values of $s_1, s_2, \cdots, s_{h-1}$ (being the assumption made in the analysis given above), the inequality of (A7.9) is the condition here for the updated value of x_h to be equal to the value of s_h regardless of what value x_h has before it is updated.

In system 9 or 11, the detected value of s_1 is taken to be the value of x_1 obtained at the end of the iterative process here. It follows that, the condition for the detected value of s_1 to be equal to s_1 (that is, the condition for error-free detection to be achieved) in the absence of noise is here given by the inequality of (A7.9) with $h = 1$, or

$$|Y_1 Z_1^T| > 2 \sum_{j=2}^{n} |Y_j Z_1^T| \qquad \text{(A7.14)}$$

which is the inequality of (A7.1).

In system 10, however, the detected value of s_1 is taken to be the value of x_1 obtained at the end of the one of the two iterative processes here associated with the smaller distance $\|R' - XY\|$, where $R' = SY$ in the absence of noise, and X (eqn. 3.13) and S (eqn. 3.5) are the two n-component vectors having the components $x_1, x_2, \cdots, x_n$ and $s_1, s_2, \cdots, s_n$ respectively. The two iterative processes of system 10 differ only in having a different value of x_1, this being such that, one of the iterative processes here is associated with $x_h = 1$ whereas the other is associated with $x_h = -1$. Clearly, if all the values of $x_2, x_3, \cdots, x_n$ obtained at the end of the iterative process associated with correct value of x_1 (that is, $x_1 = s_1$) are equal to the corresponding values of $s_2, s_3, \cdots, s_n$, then in the absence of noise, the distance $\|R' - XY\|$ or $\|SY - XY\|$ of this iterative process is equal to zero so that the value of x_1 here (being equal to s_1) is always taken as the detected value of s_1. Consequently, the condition for system 10 to achieve error-free detection in the absence of noise is when all the values of $x_2, x_3, \cdots, x_n$ obtained at the end of the iterative process associated with the correct value of x_1 ($= s_1$) are equal to the corresponding values of $s_2, s_3, \cdots, s_n$, and is therefore, from eqn. A7.9, given by

$$|Y_h Z_h^T| > 2 \sum_{j=h+1}^{n} |Y_j Z_h^T| \qquad \text{(A7.15)}$$

for $h = 2, 3, \cdots, n$ which is the inequality of (A7.2).

APPENDIX A8

DERIVATION OF THE CONDITION GIVEN IN TABLE 5.10 (CHAPTER 5)

This appendix shows that, in the detection process of system 7 (chapter 5), the necessary condition for the transmitted value of s_{i+h} $(0 \leqslant h \leqslant \lambda)$ to be included as one of the m_h' selected values for x_{i+h} (eqn. 5.54) is given by

$$|J_{i+h} + w'_{i+h}| < v_h = 1 \quad , \quad \text{if} \quad m_h' = 1 \qquad \text{(A8.1)}$$

$$|J_{i+h} + w'_{i+h}| < v_h = \sqrt{2} \quad , \quad \text{if} \quad m_h' = 2 \qquad \text{(A8.2)}$$

$$|J_{i+h} + w'_{i+h}| < v_h = 2 \quad , \quad \text{if} \quad m_h' = 4 \qquad \text{(A8.3)}$$

$$|J_{i+h} + w'_{i+h}| < v_h = 3 \quad , \quad \text{if} \quad m_h' = 9 \qquad \text{(A8.4)}$$

where the estimated value x'_{i+h} of s_{i+h} is (eqn. 5.68)

$$x'_{i+h} = s_{i+h} + J_{i+h} + w'_{i+h} \qquad \text{(A8.5)}$$

The m_h' selected values for x_{i+h} here are obtained by using the selection processes A - D described in section 5.9 (chapter 5) for $m_h' = 1$, 2, 4, and 9 respectively.

Consider first for the case when $m_h' = 1$ (selection process A). The 16-point QAM signal constellation of s_{i+h} here is divided into 16 regions shown in Fig. 5.14, and the selected value of x_{i+h} here is taken to be the possible value of s_{i+h} that lies in the same region as that of the estimate x'_{i+h} of s_{i+h}. Clearly, if x'_{i+h} lies within the circle with centre s_{i+h} and radius 1, then it can be seen from Fig. 5.14 that s_{i+h} must now lie in the same region as that of x'_{i+h}.

This means that, the value of s_{i+h} is always taken as the selected value of x_{i+h} here so long as

$$|x'_{i+h} - s_{i+h}| < 1 \qquad \text{(A8.6)}$$

or, from eqn. A8.5,

$$|J_{i+h} + w'_{i+h}| < 1 \qquad \text{(A8.7)}$$

which is the condition of (A8.1).

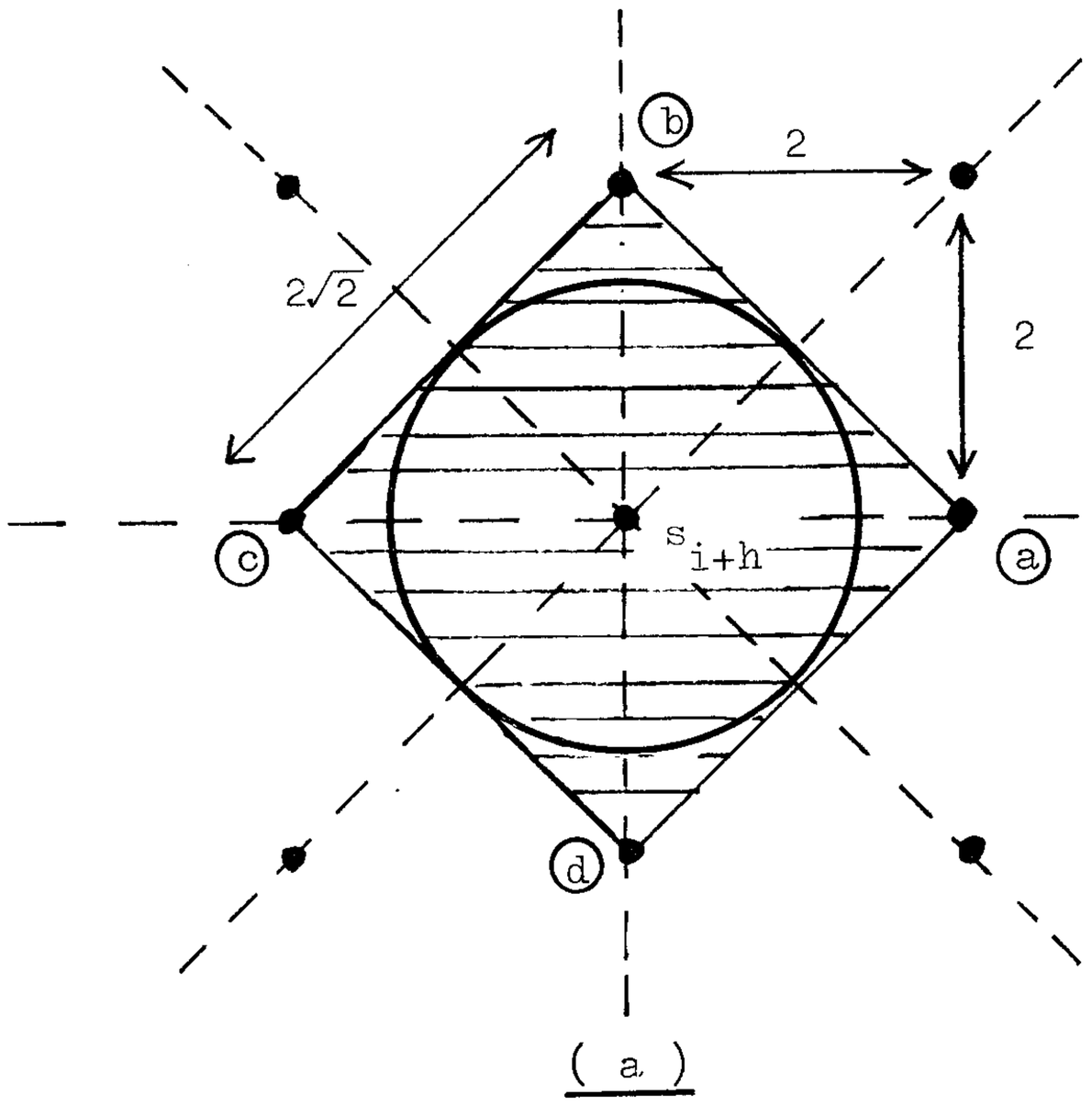
Consider next for the case when $m'_h = 2$ (selection process B). The two selected values of x_{i+h} here are taken to be the two possible values of s_{i+h} closest to the estimate x'_{i+h} of s_{i+h}. Three different situations may now arise depending on whether s_{i+h} is one of the four central points, or one of the four corner points, or one of the eight side points in the 16-point QAM signal constellation, as is shown in Fig. A8.1. It can be seen from Fig. A8.1 that, s_{i+h} is always included as one of the two selected values of x_{i+h} here so long as x'_{i+h} lies in the shaded area shown in Fig. A8.1. Furthermore, if x'_{i+h} lies in the circle with centre s_{i+h} and radius $\sqrt{2}$, then it must also lie in the shaded area here as can be seen from Fig. A8.1. It therefore follows that, s_{i+h} is always included as one of the two selected values of x_{i+h} here so long as

$$|x'_{i+h} - s_{i+h}| < \sqrt{2} \qquad \text{(A8.8)}$$

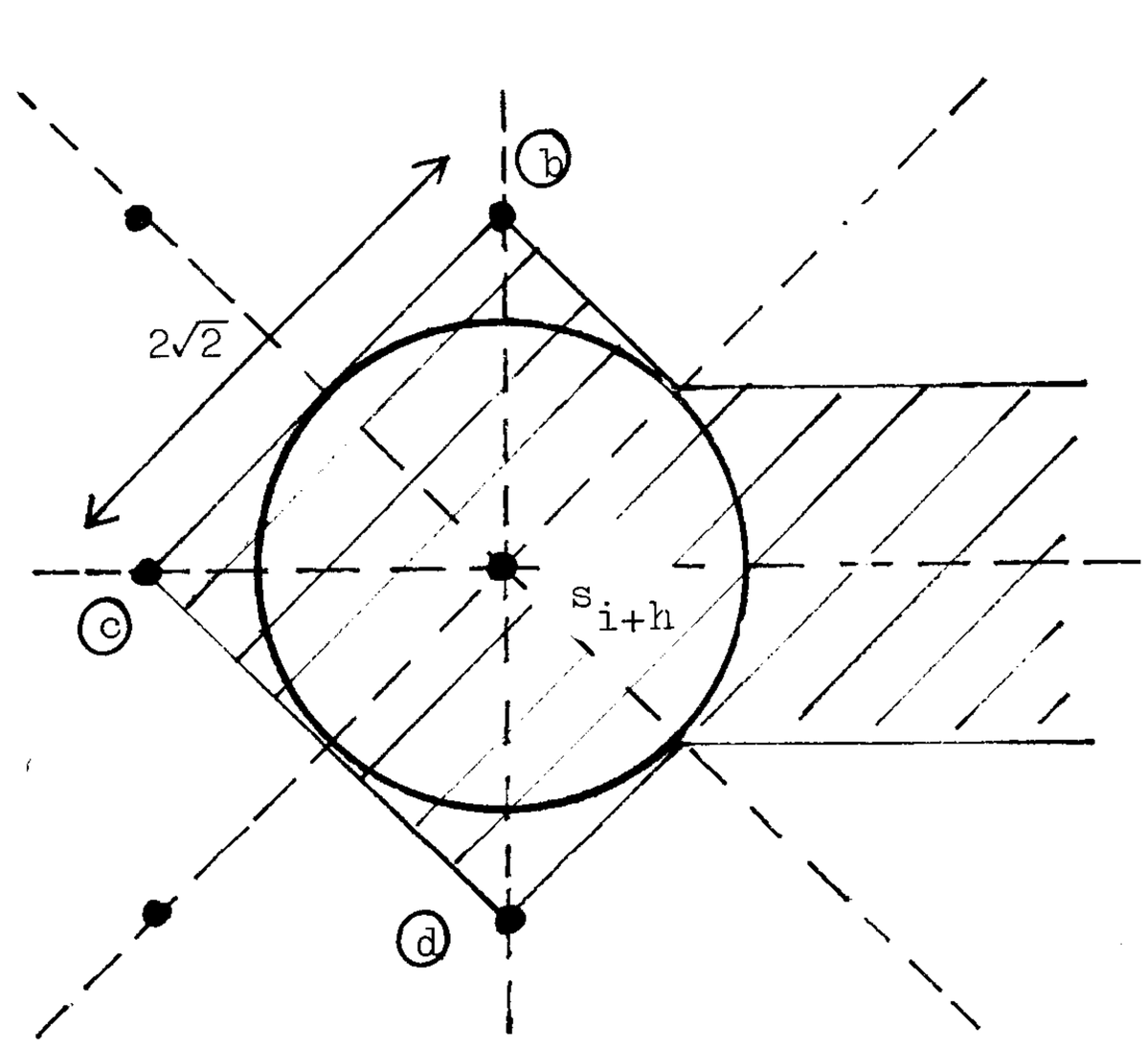
or, from eqn. A8.5,

$$|J_{i+h} + w'_{i+h}| < \sqrt{2} \qquad \text{(A8.9)}$$

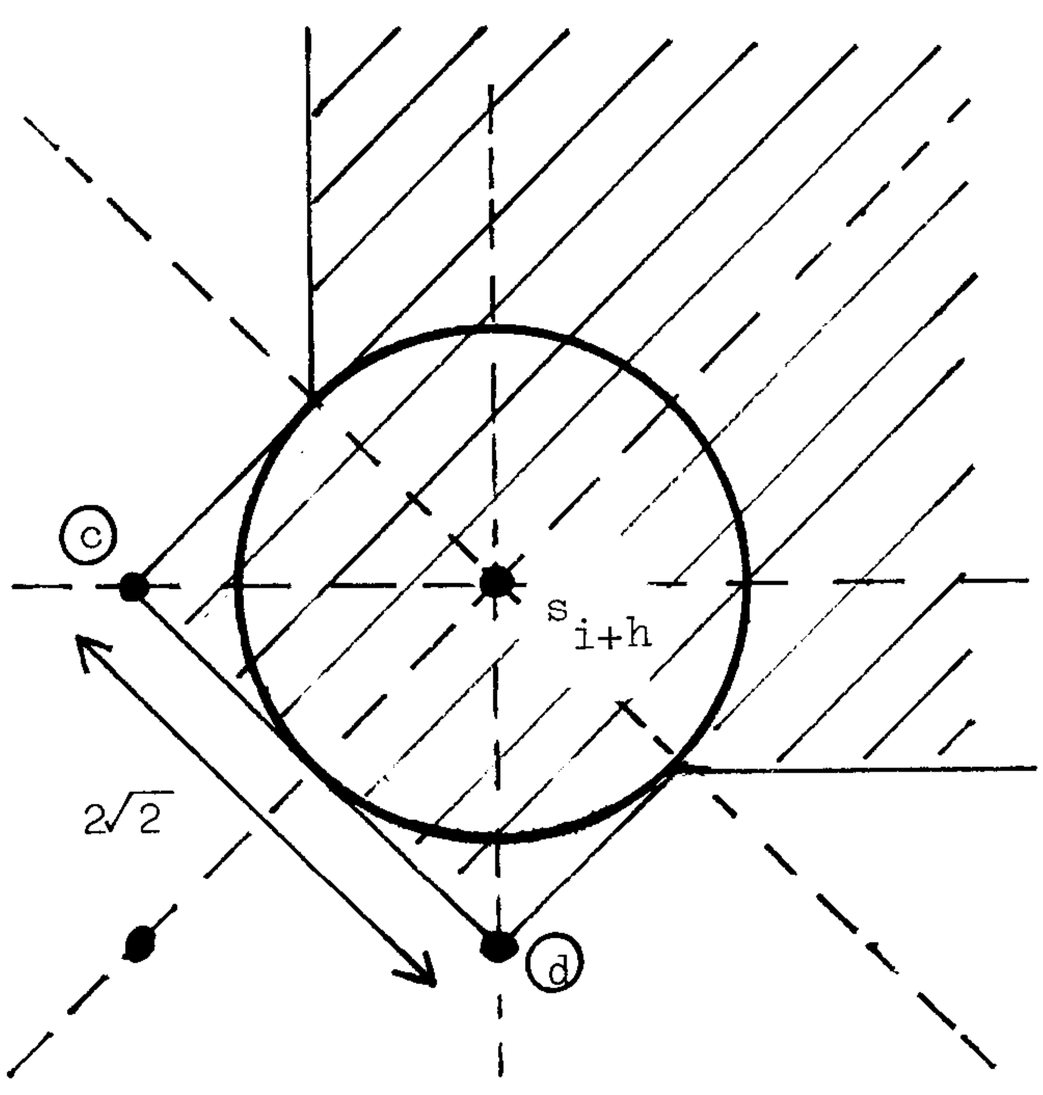
which is the condition of (A8.2).



(a)



(b)



(c)

Fig. A8.1 Part of the 16-point QAM signal constellation showing the transmitted data symbol s_{i+h} and its neighbouring points.
(a) when s_{i+h} is one of the 4 central points,
(b) when s_{i+h} is one of the 8 side points,
(c) when s_{i+h} is one of the 4 corner points.

Consider now for the case when $m'_h = 4$ (selection process C). The selection process here is such that, the 2 possible values of $Re(s_{i+h})$ that are closest to the value of $Re(x'_{i+h})$ are taken as the 2 selected values for $Re(x_{i+h})$, and the 2 possible values of $Im(s_{i+h})$ that are closest to the value of $Im(x'_{i+h})$ are taken as the 2 selected values for $Im(x_{i+h})$, to give a total of 4 selected values for x_{i+h}, where Re(.) and Im(.) are the real and imaginary parts of (.) respectively. The four possible values of $Re(s_{i+h})$ or $Im(s_{i+h})$ are here given by 3, 1, -1, and -3, and from eqn. A8.5,

$$Re(x'_{i+h}) = Re(s_{i+h}) + Re(J_{i+h} + w'_{i+h}) \qquad (A8.10)$$

and

$$Im(x'_{i+h}) = Im(s_{i+h}) + Im(J_{i+h} + w'_{i+h}) \qquad (A8.11)$$

The selection process here is also shown in Fig. 5.16. It can be seen from Fig. 5.16 that, if $Re(s_{i+h}) = 3$, then it is always included as one of the 2 selected values of $Re(x_{i+h})$ so long as $Re(x'_{i+h}) > 1$ or, from eqn. A8.10,

$$Re(J_{i+h} + w'_{i+h}) > -2 \qquad (A8.12)$$

Similarly, if $Re(s_{i+h}) = -3$, then it is always included as one of the 2 selected values of $Re(x_{i+h})$ so long as $Re(x'_{i+h}) \leq -1$ or, from eqn. A8.10,

$$Re(J_{i+h} + w'_{i+h}) \leq 2 \qquad (A8.13)$$

If $Re(s_{i+h}) = 1$, then it can be seen from Fig. 5.16 that, this value of $Re(s_{i+h})$ is always included as one of the two selected values of $Re(x_{i+h})$ so long as $Re(x'_{i+h}) > -1$ or, from eqn. A8.10,

$$Re(J_{i+h} + w'_{i+h}) > -2 \qquad (A8.14)$$

Similarly, if $\mathrm{Re}(s_{i+h}) = -1$, then it is always included as one of the two selected values of $\mathrm{Re}(x_{i+h})$ so long as $\mathrm{Re}(x'_{i+h}) \leq -1$ or, from eqn. A8.10,

$$\mathrm{Re}(J_{i+h} + w'_{i+h}) \leq 2 \qquad \text{(A8.15)}$$

Consequently, so long as $\mathrm{Re}(J_{i+h} + w'_{i+h})$ satisfies all the inequalities of (A8.12) - (A8.15), that is,

$$\left|\mathrm{Re}(J_{i+h} + w'_{i+h})\right| < 2 \qquad \text{(A8.16)}$$

then the transmitted value of $\mathrm{Re}(s_{i+h})$ is always included as one of the 2 selected values of $\mathrm{Re}(x_{i+h})$ here. The same argument can be used to show that, so long as

$$\left|\mathrm{Im}(J_{i+h} + w'_{i+h})\right| < 2 \qquad \text{(A8.17)}$$

then the transmitted value of $\mathrm{Im}(s_{i+h})$ is always included as one of the 2 selected values of $\mathrm{Im}(x_{i+h})$ here. It follows that, if both the inequalities of (A8.16) and (A8.17) are satisfied, then the transmitted value of s_{i+h} is always included as one of the 4 selected values of x_{i+h} here. Consequently, it can be seen that, the condition for the transmitted value of s_{i+h} to be included as one of the 4 selected values of x_{i+h} here is given by

$$\left|J_{i+h} + w'_{i+h}\right| < 2 \qquad \text{(A8.18)}$$

which is the same as the inequality of (A8.3).

Consider finally for the case when $m' = 9$ (selection process D). The selection process here is carried out according to Table 5.9. Thus, if $\mathrm{Re}(s_{i+h}) = \pm 1$, then it is always included as one of the 3 selected

values of $Re(x_{i+h})$ as can be seen from Table 5.9. If $Re(s_{i+h}) = 3$, then it can be seen from Table 5.9 that this value of $Re(s_{i+h})$ is always included as one of the 3 selected values of $Re(x_{i+h})$ so long as $Re(x'_{i+h}) > 0$ or, from eqn. A8.10,

$$Re(J_{i+h} + w'_{i+h}) > -3 \qquad (A8.19)$$

Similarly, if $Re(s_{i+h}) = -3$, then it is always included as one of the 3 selected values of $Re(x_{i+h})$ so long as $Re(x'_{i+h}) \leq 0$ or, from eqn. A8.10,

$$Re(J_{i+h} + w'_{i+h}) \leq 3 \qquad (A8.20)$$

Consequently, so long as $Re(J_{i+h} + w'_{i+h})$ satisfies both the inequalities of (A8.19) and (A8.20), that is,

$$|Re(J_{i+h} + w'_{i+h})| < 3 \qquad (A8.21)$$

then the transmitted value of $Re(s_{i+h})$ is always included as one of the 3 selected values of $Re(x_{i+h})$ here. The same argument can be used to show that, so long as

$$|Im(J_{i+h} + w'_{i+h})| < 3 \qquad (A8.22)$$

then the transmitted value of $Im(s_{i+h})$ is always included as one of the 3 selected values of $Im(x_{i+h})$ here. It can now be seen that, the transmitted value of s_{i+h} is always included as one of the 9 selected values of x_{i+h} here so long as

$$|J_{i+h} + w'_{i+h}| < 3 \qquad (A8.23)$$

which is the condition of (A8.4).

APPENDIX B1

COMPUTER PROGRAM FOR SYSTEM 5 (CHAPTER 3)

```
      INTEGER ERROR,CS(50),CES(50),NER(20),S(32),X(64),ES,EX(32)
      INTEGER*4 NQN
      REAL V(20),RA(32),RB(64),Y(32),C(20),D(20),U(2)
      REAL Z(40,40),ZY(40,40)
      DOUBLE PRECISION G05CBF,G05CAF,G05DDF,PP,SD
 9000 FORMAT(/////)
 9100 FORMAT(//)
 9200 FORMAT(1H ,I2,3X,I2,3X,I5,3X,I2,3X,I2)
 9300 FORMAT(1H ,F5.3,3X,I2)
 9400 FORMAT(1H ,I1)
 9500 FORMAT(1H ,F5.3)
 9600 FORMAT(1H ,'ELEMENT TRANSMITTED = ',I7)
 9700 FORMAT(1H ,'ERROR DETECTED      = ',I7)
 9800 FORMAT(1H ,'STANDARD DEVIATION  = ',F7.4)
 9900 FORMAT(1H ,'SNR IN DB           = ',F7.4)
 8400 FORMAT(1H ,8F7.3)
 8500 FORMAT(1H ,20I5)
 8600 FORMAT(1H ,'ZY-MATRIX'//)
C
C     SOME CONSTANTS
C
      READ(1,*)NG,N,NI,NB,NF
      READ(1,*)NQN
      READ(1,*)NC
      READ(1,*)(V(I),I=1,NG,1)
      READ(1,*)SD
      ERROR=0
      IB=1
      ID=1
      IE=1
      NI1=N+NI-1
      N1=N-1
      I7=NF+1
      IS=N-NF
      DO 10 I=1,20,1
      NER(I)=0
10    CONTINUE
      I1=N-NG
      DO 20 I=1,I1,1
      V(I+NG)=0.0
20    CONTINUE
C
C
C
      WRITE(1,9000)
      WRITE(1,8600)
      DO 30 I1=1,N,1
      D(I1)=0.0
      DO 40 I2=1,N,1
      Z(I1,I2)=0.0
```

```
      K1=I2-I1+1
      IF(K1.LE.0)GOTO 40
      Z(I1,I2)=V(K1)
      D(I1)=D(I1)+Z(I1,I2)*Z(I1,I2)
40    CONTINUE
30    CONTINUE
      WRITE(1,9100)
      WRITE(1,8400)(D(I),I=1,N,1)
      WRITE(1,9100)
      WRITE(1,8600)
      DO 50 I=1,N,1
      DO 60 J=1,N,1
      ZY(I,J)=0.0
      DO 70 K=1,N,1
      K1=K-J+1
      IF(K1.LE.0)GOTO 70
      ZY(I,J)=ZY(I,J)+Z(I,K)*V(K1)
70    CONTINUE
60    CONTINUE
      WRITE(1,8400)(ZY(I,L),L=1,N,1)
50    CONTINUE
C
C     STARTING PROCEDURE
C
888   CALL G05CBF(NQN)
999   DO 80 I=1,N,1
      S(I)=1
      EX(I)=1
      RA(I)=G05DDF(0.000,50)
80    CONTINUE
      ES=1
C
C     STORING THRESHOLD VALUES
C
      DO 90 I=1,N,1
      Y(I)=0.0
      DO 100 J=1,N,1
      Y(I)=Y(I) + Z(I,J)*Z(I,J)
100   CONTINUE
90    CONTINUE
C
C
C     DETECTION PROCESS
C
C
      IA=1
C
C
C
777   IS=S(1)
```

```
      DO 110 I=1,N1,1
      K1=I+1
      S(I)=S(K1)
      EX(I)=EX(K1)
      RA(I)=RA(K1)
110   CONTINUE
C
C     GENERATING NEXT ELEMENT
C
      SN=G05CAF(PP)
      IF(SN.GT.0.5)GOTO 120
      S(N)=-1
      GOTO 130
120   S(N)=+1
130   RA(N)=G05DDF(0.000,SD)
      DO 140 K=2,NG,1
      K1=1+N-K
      IF(K1.LE.0)GOTO 140
      K2=S(K1)-EX(K1)
      VK1=V(K)+V(K)
      IF(K2.EQ.  2)RA(N)=RA(N)+VK1
      IF(K2.EQ.-2)RA(N)=RA(N)-VK1
140   CONTINUE
      IF(S(N).EQ.  1)RA(N)=RA(N)+V(1)
      IF(S(N).EQ.-1)RA(N)=RA(N)-V(1)
C
C     ITERATIVE PROCESS TO DETERMINE S FROM RA
C
      EX(N)=0
      X(I8)  =+1
      X(I8+N)=-1
      DO 150 JJ=1,2,1
      U(JJ)=0.0
      IF(JJ.EQ.1)NN=0
      IF(JJ.EQ.2)NN=N
      DO 160 I=1,N,1
      L=I+NN
      RB(L)=RA(I)
      IF(I.EQ.I8)GOTO 160
      X(L)=EX(I)
160   CONTINUE
C
      K1=EX(I8)
      K2=X(I8+NN)
      K3=0
      DO 165 K=I8,N,1
      K3=K3+1
      K4=K+NN
      IF(K1.EQ.  1)RB(K4)=RB(K4)+V(K3)
      IF(K1.EQ.-1)RB(K4)=RB(K4)-V(K3)
```

```
      IF(K2.EQ.  1)RB(K4)=RB(K4)-V(K3)
      IF(K2.EQ. -1)RB(K4)=RB(K4)+V(K3)
165   CONTINUE
C
      IF(NF.LE.0)GOTO 170
      DO 180 I=1,NF,1
      J=N-I+1
      JX=X(J+NN)
      K1=0
      DO 185 K=J,N,1
      K1=K1+1
      K2=K+NN
      IF(JX.EQ.  1)RB(K2)=RB(K2)+V(K1)
      IF(JX.EQ. -1)RB(K2)=RB(K2)-V(K1)
185   CONTINUE
      X(J+NN)=0
180   CONTINUE
170   IC=1
666   DO 190 I=1,N,1
      L=I
      EL=0.0
      L1=L+NN
      LX=X(L1)
      YL=Y(L)
      IF(IC.EQ.1.AND.LX.NE.0)GOTO 190
      DO 200 K=1,N,1
      K1=K+NN
      EL=EL+RB(K1)*Z(L,K)
200   CONTINUE
      IX=0
210   IF(LX)220,240,230
220   IF(EL.GT.YL)IX=2
      GOTO 250
230   IF(EL.LT.-YL)IX=-2
      GOTO 250
240   IF(EL)260,260,270
260   IX=-1
      GOTO 250
270   IX=+1
250   IF(IX.EQ.0)GOTO 190
      DO 280 J=1,N,1
      K1=J-L+1
      IF(K1.LE.0)GOTO 280
      VK1=V(K1)
      J1=J+NN
      IF(IX.GE.  2)VK1=VK1+VK1
      IF(IX.LE. -2)VK1=VK1+VK1
      IF(IX.LT.0)RB(J1)=RB(J1)+VK1
      IF(IX.GT.0)RB(J1)=RB(J1)-VK1
280   CONTINUE
```

```
      X(L1)=I X+IX
190   CONTINUE
      IF(IC.GE.NC)GOTO 290
      IC=IC+1
      GOTO 666
290   DO 300 J=1,N,1
      J1=J+NN
      U(JJ)=U(JJ)+RB(J1)*RB(J1)
300   CONTINUE
150   CONTINUE
      IF(U(1).GT.U(2))GOTO 310
      ES=X(1)
      DO 320 I=1,N,1
      EX(I)=X(I)
      RA(I)=RB(I)
320   CONTINUE
      GOTO 330
310   ES=X(1+N)
      DO 340 I=1,N,1
      L=I+N
      EX(I)=X(L)
      RA(I)=RB(L)
340   CONTINUE
330   IC=1
      IF(IA.LT.N)GOTO 350
      IF(S(1).EQ.ES)GOTO 360
      ERROR=ERROR+1
      NER(IE)=NER(IE)+1
360   IF(ID.LT.50)GOTO 370
      IF(IE.LT.20)GOTO 380
      WRITE(1,8500)(NER(I),I=1,20,1)
      DO 390 I=1,20,1
      NER(I)=0
390   CONTINUE
      IE=0
380   IE=IE+1
      ID=0
370   ID=ID+1
350   IF(IA.GE.NI1)GOTO 400
      IA=IA+1
      GOTO 777
400   IF(SD.LE.0.0)GOTO 410
      SNR=10.0*(ALOG10(1.0/SD/SD))
410   WRITE(1,9000)
      WRITE(1,9200)NG,N,NI,NB,NF
      WRITE(1,9300)SD,NQN
      WRITE(1,9400)NC
      WRITE(1,9500)(V(I),I=1,NG,1)
      WRITE(1,9000)
      WRITE(1,9600)NI
```

```
      WRITE(1,9700)ERROR
      WRITE(1,9500)SD
      WRITE(1,9900)SNR
      WRITE(1,9000)
      WRITE(1,9100)
      IF(IB.GE.NB)GOTO 420
      IB=IB+1
      IF(IB.EQ.6)READ(1,*)NON
      READ(1,*)SD
      ERROR=0
      IC=1
      ID=1
      IE=1
      IF(IB.EQ.6)GOTO 999
      GOTO 999
420   CALL EXIT
      END
```

APPENDIX B2

COMPUTER PROGRAM FOR SYSTEM 8 WITH VERSION b (CHAPTER 5)

```
      PROGRAM QAM(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT)
C
C
C
      INTEGER  RS(65),IS(65),RES,IES,RXS(40,40),IXS(40,40),IY(40)
     1,ERRORB,ERRORS,NERB(20),NERS(20),NEIB(20),NEIS(20)
     2,RXA(600),IXA(600),RYS(40,40),IYS(40,40),RXN(20),IXN(20)
     3,RZS(16),IZS(16),RDIF,IDIF
      REAL RV(50),IV(50),MINU,RSUM(9),ISUM(9),RZUM(9),IZUM(9),RYUM,IYUM,
     1 RR(65),IR(65),U(600),YU(40),D(600),YD(40),TH(16),
     2 RXUM,IXUM
9000  FORMAT(/////)
9101  FORMAT(1H0,'BIT      TRANSMITTED = ',I7)
9102  FORMAT(1H0,'SYMBOL   TRANSMITTED = ',I7)
9201  FORMAT(1H0,'BIT      ERROR       = ',I7)
9202  FORMAT(1H0,'SYMBOL   ERROR       = ',I7)
9300  FORMAT(1H0,'MEAN TXSIGNAL POWER = ',F7.4)
9400  FORMAT(1H0,'STANDARD DEVIATION  = ',F7.4)
9500  FORMAT(1H0,'SNR IN DB           = ',F7.4)
9999  FORMAT(1H1)
8200  FORMAT(//1H1,'NOISE - 0 ')
8600  FORMAT(//)
8700  FORMAT(1H ,32I3)
8800  FORMAT(1H+,98X,F10.4,2X,F10.4)
8900  FORMAT(/)
7000  FORMAT(1H ,4(I2,3X),I5,3X,5(I2,3X))
7100  FORMAT(1H ,F6.4,3X,I2)
7200  FORMAT(1H ,6(/10F7.4))
7300  FORMAT(1H 16I3)
7400  FORMAT(1H 20I5)
7500  FORMAT(1H ,'U=',F12.5,5X,'YD=',F12.5)
C
C     SOME CONSTANTS
C
      READ(1,*)NG,NM,NL,N,NI,NB,NF,NF1,N2
      READ(1,*)NON
      READ(1,*)(RV(I),I=1,NG,1)
      READ(1,*)(IV(I),I=1,NG,1)
      READ(1,*)SD
      N=N+1
      ERRORB=0
      ERRORS=0
      IB=1
      IC=1
      ID=1
      IH=0
      NM1=NM-NL-1
      NI1=2*NI
```

```
      NI2=2*NI1
      IF(N.GE.NG)GOTO 10
      IH=NG-N
10    DO 20 I=1,20,1
      NERB(I)=0
      NERS(I)=0
      NEIB(I)=0
      NEIS(I)=0
20    CONTINUE
      N1=N+IH+NF+3
      N4=NM*N2
      N5=N4+1
      N7=N4-NM
      I1=NI+N-1
      I2=N1-1
      I3=N-1
      I7=NF+1
      I7A=I7+1
      I71=NF1+1
      I71A=I71+1
      I8=NF+3
      NN=I3+3
      BB=0.0
      DO 30 I=1,I7,1
      BB=BB + (RV(I)*RV(I)) + (IV(I)*IV(I))
30    CONTINUE
      AA=BB + (RV(I7A)*RV(I7A)) + (IV(I7A)*IV(I7A))
      A2R=0.0
      A2I=0.0
      DO 40 I=1,I7,1
      VR1=+RV(I)
      VI1=+IV(I)
      VR2=+RV(I+1)
      VI2=-IV(I+1)
      A2R=A2R + (VR1*VR2) - (VI1*VI2)
      A2I=A2I + (VR1*VI2) + (VI1*VR2)
40    CONTINUE
      ALPHA=(AA*BB) - (A2R*A2R + A2I*A2I)
      TH(1)=1.0/16.0
      DO 50 I=1,14,1
      TH(I+1)=TH(I)+TH(1)
50    CONTINUE
      K1=-2
      K3=-4
      DO 60 I=1,4,1
      K2=-2
      K1=K1+2
      K3=K3+4
      DO 70 J=1,4,1
      K2=K2+2
```

```
      K4=K3+J
      RXN(K4)=-3+K1
      IXN(K4)=-3+K2
70    CONTINIE
60    CONTINUE
C
C     STARTING PROCEDURE
C
      CALL G05CGF(NON)
999   WRITE(2,9999)
      DO 80 I=1,N1,1
      RS(I)=1
      IS(I)=1
      RR(I)=G05DDF(0.0,SD)
      IR(I)=G05DDF(0.0,SD)
      L1=I
      IF(I.GT.NG)L1=NG
      DO 90 J=1,L1,1
      VR=RV(J)
      VI=IV(J)
      RR(I)=RR(I) + (VR-VI)
      IR(I)=IR(I) + (VI+VR)
90    CONTINUE
80    CONTINUE
      TXPOW=0.0
C
C     STORING VECTORS
C
      DO 100 I=1,NM,1
      U(I)=10E10
      DO 110 J=1,I3,1
      RXS(I,J)=1
      IXS(I,J)=1
110   CONTINUE
100   CONTINUE
      U(I)=0.0
      DO 120 J=1,I3,1
      K1=NN+1
      RXS(K1,J)=18
      IXS(K1,J)=18
120   CONTINUE
C
C     GENERATING NEXT ELEMENT
C
      RES=1
      IES=1
      DO 888 IA=1,I1,1
      DO 130  I=1,I2,1
      K1=I+1
      RS(I)=RS(K1)
```

```
      IS(I)=IS(K1)
      RR(I)=RR(K1)
      IR(I)=IR(K1)
130   CONTINUE
      SN=G05CAF(PP)
      K2=1
      DO 140 I=1,15,1
      IF(SN.LT.TH(I))GOTO 150
      K2=I+1
140   CONTINUE
150   RS(N1)=RXN(K2)
      IS(N1)=IXN(K2)
      RR(N1)=G05DDF(0.0,SD)
      IR(N1)=G05DDF(0.0,SD)
      DO 160 J=1,NG,1
      K1=N1-J+1
      SR=RS(K1)
      SI=IS(K1)
      VR=RV(J)
      VI=IV(J)
      RR(N1)=RR(N1) + (SR*VR - SI*VI)
      IR(N1)=IR(N1) + (SR*VI + SI*VR)
160   CONTINUE
      IF(IH.LE.0)GOTO 170
C
C     INTERSYMBOL INTERFERENCE CANCELATION CIRCUIT
C
      DO 180 J=1,IH,1
      L7=I3+J
      L8=N+J
      VR=RV(L8)
      VI=IV(L8)
      RR(L7)=RR(L7) - (RES*VR - IES*VI)
      IR(L7)=IR(L7) - (RES*VI + IES*VR)
180   CONTINUE
C
C     DERIVING  (NM*N2)  VECTORS
C     AND
C     CALCULATING NEW COSTS U AND D
C
170   DO 190 I=1,NM,1
      K1=I+N7
      DO 200 J=I,K1,NM
      D(J)=U(I)
200   CONTINUE
      DO 210 J=1,I8,1
      RSUM(J)=0.0
      ISUM(J)=0.0
      L3=I3+J
      DO 220 K=1,I3,1
```

```
      L4=I3-K+1
      L5=J+K
      KR=RXS(I,L4)
      KI=IXS(I,L4)
      IF(KR.GT.L6)GOTO 220
      IF(KI.GT.L6)GOTO 220
      VR=RV(L5)
      VI=IV(L5)
      RSUM(J)=RSUM(J) + (KR*VR - KI*VI)
      ISUM(J)=ISUM(J) + (KR*VI + KI*VR)
220   CONTINUE
      RSUM(J)=RR(L3) - RSUM(J)
      ISUM(J)=IR(L3) - ISUM(J)
210   CONTINUE
C
C     FIRST EXPANSION OF VECTORS
C
      IF(N2.EQ.16)GOTO 230
      RXUM=0.0
      IXUM=0.0
      RYUM=0.0
      IYUM=0.0
      DO 240 KA=1,17,1
      KB=KA
      KC=KA+1
      ZR=RSUM(KC)
      ZI=ISUM(KC)
      VRA=+RV(KC)
      VIA=-IV(KC)
      VRB=+RV(KB)
      VIB=-IV(KB)
      RXUM=RXUM + (ZR*VRA - ZI*VIA)
      IXUM=IXUM + (ZR*VIA + ZI*VRA)
      RYUM=RYUM + (ZR*VRB - ZI*VIB)
      IYUM=IYUM + (ZR*VIB + ZI*VRB)
240   CONTINUE
      ZR=RSUM(1)
      ZI=ISUM(1)
      VRA=+RV(1)
      VIA=-IV(1)
      RXUM=RXUM + (ZR*VRA - ZI*VIA)
      IXUM=IXUM + (ZR*VIA + ZI*VRA)
      XAR=((RXUM*BB) - (RYUM*A2R) + (IYUM*A2I))/ALPHA
      XAI=((IXUM*BB) - (RYUM*A2I) - (IYUM*A2R))/ALPHA
      IF(N2.EQ.9)GOTO 250
      KA=-5
      IF(XAR.GT.-1.0)KA=-3
      IF(XAR.GT.+1.0)KA=-1
      KB=-5
      IF(XAI.GT.-1.0)KB=-3
```

```
      IF(XAI.GT.+1.0)KB=-1
      KE=0
      DO 260 KC=1,2,1
      KA=KA+2
      KF=KB
      DO 270 KD=1,2,1
      KE=KE+1
      KF=KF+2
      RZS(KE)=KA
      IZS(KE)=KF
270   CONTINUE
260   CONTINUE
      GOTO 280
250   KA=-5
      IF(XAR.GT.0.0)KA=-3
      KB=-5
      IF(XAI.GT.0.0)KB=-3
      KE=0
      DO 290 KC=1,3,1
      KA=KA+2
      KF=KB
      DO 300 KD=1,3,1
      KE=KE+1
      KF=KF+2
      RZS(KE)=KA
      IZS(KE)=KF
300   CONTINUE
290   CONTINUE
      GOTO 280
230   DO 310 K=1,N2,1
      RZS(K)=RXN(K)
      IZS(K)=IXN(K)
310   CONTINUE
C
C     SECOND EXPANSION OF VECTORS
C
280   I6=I-NM
      DO 320 K=1,N2,1
      I6=I6+NM
      KR=RZS(K)
      KI=IZS(K)
      RXA(I6)=KR
      IXA(I6)=KI
      DO 330 KA=1,I8,1
      VR=RV(KA)
      VI=IV(KA)
      RZUM(KA)=RSUM(KA) - (KR*VR - KI*VI)
      IZUM(KA)=ISUM(KA) - (KR*VI + KI*VR)
330   CONTINUE
      RXUM=0.0
```

```
      IXUM=0. 0
      RYUM=0. 0
      IYUM=0. 0
      DO 340 KA=1. I7. 1
      KB=KA+1
      KC=KA+2
      ZR=RZUM(KC)
      ZI=IZUM(KC)
      VRA=+RV(KB)
      VIA=-IV(KB)
      VRB=+RV(KA)
      VIB=-IV(KA)
      RXUM=RXUM + (ZR*VRA - ZI*VIA)
      IXUM=IXUM + (ZR*VIA + ZI*VRA)
      RYUM=RYUM + (ZR*VRB - ZI*VIB)
      IYUM=IYUM + (ZR*VIB + ZI*VRB)
  340 CONTINUE
      ZR=RZUM(2)
      ZI=IZUM(2)
      VRA=+RV(1)
      VIA=-IV(1)
      RXUM=RXUM + (ZR*VRA - ZI*VIA)
      IXUM=IXUM + (ZR*VIA - ZI*VRA)
      XAR=((RXUM*BB) - (RYUM*A2R) + (IYUM*A2I))/ALPHA
      XAI=((IXUM*BB) - (RYUM*A2I) - (IYUM*A2R))/ALPHA
      IF(XAR. GT. +3. 0)XAR=+3. 0
      IF(XAR. LT. -3. 0)XAR=-3. 0
      IF(XAI. GT. +3. 0)XAI=+3. 0
      IF(XAI. LT. -3. 0)XAI=-3. 0
      XBR=(RYUM - (XAR*A2R) - (XAI*A2I))/BB
      XBI=(IYUM + (XAR*A2I) - (XAI*A2R))/BB
      IF(XBR. GT. +3. 0)XBR=+3. 0
      IF(XBR. LT. -3. 0)XBR=-3. 0
      IF(XBI. GT. +3. 0)XBI=+3. 0
      IF(XBI. LT. -3. 0)XBI=-3. 0
      ZR=RZUM(1)
      ZI=IZUM(1)
      D(I6)=D(I6) + (ZR*ZR) + (ZI*ZI)
      U(I6)=D(I6)
      DO 350 KA=1. 2. 1
      XR=XAR
      XI=XAI
      IF(KA. EQ. 2)XR=XBR
      IF(KA. EQ. 2)XI=XBI
      KB=KA+1
      KD=0
      DO 360 KC=KB. I8. 1
      KD=KD+1
      VR=RV(KD)
      VI=IV(KD)
```

```
      RZUM(KC)=RZUM(KC) - (XR*VR - XI*VI)
      IZUM(KC)=IZUM(KC) - (XR*VI + XI*VR)
360   CONTINUE
350   CONTINUE
      DO 370 KA=1,I71A,1
      KB=KA+1
      ZR=RZUM(KB)
      ZI=IZUM(KB)
      D(I6)=D(I6) + (ZR*ZR) + (ZI*ZI)
370   CONTINUE
320   CONTINUE
190   CONTINUE
C
C     DETECT S(1) AS X(1)
C
      DO 380 I=1,NM,1
      IY(I)=N5
      YD(I)=10E9
380   CONTINUE
      DO 390 I=1,N4,1
      IF(D(I).GT.YD(1))GOTO 390
      IY(1)=I
      YD(1)=D(I)
390   CONTINUE
      MINI=IY(1)
400   IF(MINI.LE.NM)GOTO 410
      MINI=MINI-NM
      GOTO 400
410   RES=RXS(MIN,1)
      IES=IXS(MINI,1)
      D(IY(1))=10E10
C
C     SELECTING NL BEST PREVIOUS VECTORS
C     SELECTING NM1 BEST PRESENT VECTORS
C     (ALL WITH ANTI-MERGING)
C
      L1=1
      IF(NL.LE.0)GOTO 420
C
C     SELECTING NL BEST PREVIOUS VECTORS
C
      DO 430 I=1,NL,1
      IF(RXS(I,1).NE.RES)GOTO 430
      IF(IXS(I,1).NE.IES)GOTO 430
      L1=L1+1
      K1=I+N7
      DO 440 J1=I,K1,NM
      IF(D(J1).GT.YD(L1))GOTO 440
      IY(L1)=J1
      YD(L1)=D(J1)
```

```
  440  CONTINUE
       D(IY(L1))=10E10
  430  CONTINUE
C
C      SELECTING NM1 BEST PRESENT VECTORS
C
  420  DO 450 I=1,NM,1
       IF(RXS(I,1).NE.RES)GOTO 450
       IF(IXS(I,1).NE.IES)GOTO 450
       K1=I+N7
       DO 460 J1=I,K1,NM
       K2=NM
       DO 470 K=1,NM1,1
       IF(D(J1).GT.YD(K2))GOTO 480
       K2=K2-1
  470  CONTINUE
  480  IF(K2.GE.NM)GOTO 460
       K2=K2+2
       IF(K2.GT.NM)GOTO 490
       K4=NM
       DO 500 K3=K2,NM,1
       K5=K4
       K4=K4-1
       IY(K5)=IY(K4)
       YD(K5)=YD(K4)
  500  CONTINUE
  490  K2=K2-1
       IY(K2)=J1
       YD(K2)=D(J1)
  460  CONTINUE
  450  CONTINUE
C
C      SETTING VALUES FOR YU AND YS
C
       DO 510 I=1,NM,1
       MINI=IY(I)
       IF(MINI.LT.NS)GOTO 520
       DO 530 J=1,N,1
       RYS(I,J)=10
       IYS(I,J)=10
  530  CONTINUE
       YU(I)=10E10
       GOTO 510
  520  YU(I)=U(MINI)
       RYS(I,N)=RXA(MINI)
       IYS(I,N)=IXA(MINI)
  540  IF(MINI.LE.NM)GOTO 550
       MINI=MINI-NM
       GOTO 540
  550  DO 560 J=1,I3,1
```

```
      RYS(I,J)=RXS(MINI,J)
      IYS(I,J)=IXS(MINI,J)
560   CONTINUE
510   CONTINUE
C
C     TRANSFER Y BACK TO X
C
      DO 570 I=1,NM,1
      U(I)=YU(I)
      DO 580 J=1,I3,1
      K1=J+1
      RXS(I,J)=RYS(I,K1)
      IXS(I,J)=IYS(I,K1)
580   CONTINUE
570   CONTINUE
C
C     CALCULATE ERROR RATE
C
      IF(IA.LT.I3)GOTO 888
      TXPOW=TXPOW + (RS(1)*RS(1) + (IS(1)*IS(1))
      RDIF=RES-RS(1)
      IDIF=IES-IS(1)
      IF(RDIF.LT.0)RDIF=-RDIF
      IF(IDIF.LT.0)IDIF=-IDIF
      IF(RDIF.EQ.0)GOTO 590
      NERS(ID)=NERS(ID) + 1
      ERRORS=ERRORS + 1
      IF(RDIF.NE.4)GOTO 600
      NERB(ID)=NERB(ID) + 2
      ERRORB=ERRORB + 2
      GOTO 590
600   NERB(ID)=NERB(ID) + 1
      ERRORB=ERRORB + 1
590   IF(IDIF.EQ.0)GOTO 610
      NEIS(ID)=NEIS(ID) + 1
      ERRORS=ERRORS + 1
      IF(IDIF.NE.4)GOTO 620
      NEIB(ID)=NEIB(ID) + 2
      ERRORB=ERRORB + 2
      GOTO 610
620   NEIB(ID)=NEIB(ID) + 1
      ERRORB=ERRORB + 1
610   IF(IC.LT.50)GOTO 630
      IF(ID.LT.20)GOTO 640
      WRITE(2,7400)(NERB(I),I=1,20,1)
      WRITE(2,7400)(NEIB(I),I=1,20,1)
      WRITE(2,7400)(NERS(I),I=1,20,1)
      WRITE( 2,7400)(NEIS(I),I=1,20,1)
      DO 650 I=1,20,1
      NERB(I)=0
```

```
      NEIB(I)=0
      NERS(I)=0
      NEIS(I)=0
650   CONTINUE
      ID=1
      GOTO 660
640   ID=ID+1
660   IC=1
      GOTO 888
630   IC=IC+1
888   CONTINUE
C
C     CALCULATE SNR IN DB
C
      TXPOW=TXPOW/(NI+1.0)
      IF(SD.LE.0.0)GOTO 670
      SNR=10.0*ALOG10(TXPOW/SD/SD)
      GOTO 680
670   WRITE(2,8200)
680   WRITE(2,9000)
      WRITE(2,8700)(RS(J),J=2,N,1)
      WRITE(2,8700)(IS(J),J=2,N,1)
      WRITE(2,8900)
      DO 690 I=1,NM,1
      WRITE(2,8700)(RXS(I,J),J=1,I3,1)
      WRITE(2,8700)(IXS(I,J),J=1,I3,1)
      WRITE(2,8800)U(I),YD(I)
690   CONTINUE
      WRITE(2,9000)
      WRITE(2,7000)NG,NM,NL,I3,NI,NB,NF,NF1,N2
      WRITE(2,7100)SD,NON
      WRITE(2,7200)(RV(I),I=1,NG,1)
      WRITE(2,7200)(IV(I),I=1,NG,1)
      WRITE(2,9000)
      WRITE(2,9101)NI2
      WRITE(2,9102)NI1
      WRITE(2,9201)ERRORB
      WRITE(2,9202)ERRORS
      WRITE(2,9300)TXPOW
      WRITE(2,9400)SD
      WRITE(2,9500)SNR
      IF(IB.GE.NB)GOTO 700
      ERRORB=0
      ERRORS=0
      READ(1,*)SD
      IB=IB+1
      GOTO 999
700   STOP
      END
```

APPENDIX B3

COMPUTER PROGRAM FOR CODED SYSTEM WITH CODER 1 (CHAPTER 7)

```
      PROGRAM COD(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT)
C
C
C
C
      INTEGER S(130),ES1,ES2,ES3,ES4,ES5,ES6,XS(70,130),G(2,10),P(2),
     1 ERROR1,ERROR2,ERROR3,ERROR4,ERROR5,ERROR6,YS(50,130),NXS(130),
     2 SS(130),CS1(50),CS2(50),CS3(50),CS4(50),CS5(50),CS6(50),CES1(50)
     3 CES2(50),CES3(50),CES4(50),CES5(50),CES6(50),NER1(20),NER2(20),
     4 NER3(20),NER4(20),NER5(20),NER6(50)
      REAL V(20),U(70),MINU,MINV,YU(50),Q(130),XQ(70,130),Z(70),
     1 YQ(50,130),R(130),C01(50),C02(50),C03(50),C04(50),C05(50),C06(50
9000  FORMAT(/////)
9100  FORMAT(1H0,'ELEMENT TRANSMITTED = ',I7)
9201  FORMAT(1H0,'ERROR  (N=128)      = ',I7)
9202  FORMAT(1H0,'ERROR  (N=96)       = ',I7)
9203  FORMAT(1H0,'ERROR  (N=64)       = ',I7)
9204  FORMAT(1H0,'ERROR  (N=48)       = ',I7)
9205  FORMAT(1H0,'ERROR  (N=32)       = ',I7)
9206  FORMAT(1H0,'ERROR  (N=16)       = ',I7)
9300  FORMAT(1H0,'MEAN TXSIGNAL POWER = ',F7.4)
9400  FORMAT(1H0,'STANDARD DEVIATION  = ',F7.4)
9500  FORMAT(1H0,'SNR IN DB           = ',F7.4)
9999  FORMAT(1H1)
8000  FORMAT(//1H0,(F6.3))
8100  FORMAT(1H ,'Q = ',16F5.2)
8200  FORMAT(//1H0,'NOISE - 0 ')
8300  FORMAT(//)
8400  FORMAT(1H ,32I2)
8500  FORMAT(1H ,70X,F12.5,5X,F12.5)
7000  FORMAT(1H ,I1,3X,I3,3X,I2,3X,I3,3X,I5,3X,I1,3X,I1,3X,I1)
7100  FORMAT(1H ,F5.3,3X,I2)
7200  FORMAT(1H ,F5.3)
7300  FORMAT(1H ,8I1)
7400  FORMAT(1H ,20I5)
C
C     SOME CONSTANTS
C
      READ(1,*)NG,NM,NL,N,NI,NB,NH,NK
      READ(1,*)NQN
      READ(1,*)(V(I),I=1,NG,1)
      READ(1,*)((G(I,J),J=1,NK,1),I=1,NH,1)
      READ(1,*)SD
      EQ=0.0
      ERROR1=0
      ERROR2=0
      ERROR3=0
      ERROR4=0
      ERROR5=0
```

```
      ERROR6=0
      IB=1
      IC=1
      IO=1
      DO 10 I=1,2,1
      U(2*NM+I)=10E10
      DO 20 J=1,N,1
      XS(2*NM+I,J)=0
      XO(2*NM+I,J)=0.0
20    CONTINUE
10    CONTINUE
      DO 30 I=1,20,1
      NER1(I)=0
      NER2(I)=0
      NER3(I)=0
      NER4(I)=0
      NER5(I)=0
      NER6(I)=0
30    CONTINUE
C
C     STARTING PROCEDURE
C
      CALL G05CBF(NON)
999   WRITE(2,9999)
      DO 40 I=1,N,1
      S(I)=+1
      Q(I)=0.0
      R(I)=0.0
40    CONTINUE
      DO 50 I=1,N,1
      J1=N-1
      DO 60 J=1,J1,1
      S(J)=S(J+1)
      Q(J)=Q(J+1)
      R(J)=R(J+1)
60    CONTINUE
      S(N)=+1
      DO 70 K=1,N,1
      SS(K)=S(K)
70    CONTINUE
      CALL ENCO(SS,G,QQ,N,NH,NK)
      Q(N)=QQ
      R(N)=G05ODF(0.0,50)
      DO 80 K=1,NG,1
      R(N)=R(N) + Q(1+N-K)*V(K)
80    CONTINUE
50    CONTINUE
C
C     STORING VECTORS
C
```

```
      DO 90 I=1,NM,1
      U(I)=10E10
      DO 100 J=1,N,1
      XS(I,J)=+1
      XQ(I,J)=Q(J)
100   CONTINUE
90    CONTINUE
      U(1)=0.0
C
C     GENERATING NEXT ELEMENT
C
      J3=NI+N-1
      DO 888 IA=1,J3,1
      J4=N-1
      DO 110 I=1,J4,1
      S(I)=S(I+1)
      Q(I)=Q(I+1)
      R(I)=R(I+1)
110   CONTINUE
      SN=G05CAF(PP)
      IF(SN.GT.0.5)GOTO 120
      S(N)=-1
      GOTO 130
120   S(N)=+1
130   DO 140 K=1,N,1
      SS(K)=S(K)
140   CONTINUE
      CALL ENCO(SS,G,QQ,N,NH,NK)
      Q(N)=QQ
      R(N)=G05DDF(0.0,0.5D)
      DO 150 K=1,NG,1
      R(N)=R(N) + Q(1+N-K)*V(K)
150   CONTINUE
C
C     EXPANDING STORED VECTORS
C
      DO 160 I=1,NM,1
      J5=N-1
      DO 170 J=1,J5,1
      XS(I,J)=XS(I,J+1)
      XS((I+NM),J)=XS(I,J+1)
      XQ(I,J)=XQ(I,J+1)
      XQ((I+NM),J)=XQ(I,J+1)
170   CONTINUE
160   CONTINUE
      DO 180 I=1,NM,1
      XS(I,N)=+1
      K3=I+NM
      XS(K3,N)=-1
180   CONTINUE
```

```
      JE=NM*2
      DO 190 I=1,JE,1
      DO 200 J=1,N,1
      NXS(J)=XS(I,J)
200   CONTINUE
      CALL ENCO(NXS,G,QQ,K4,NH,NK)
      XQ(I,N)=QQ
190   CONTINUE
C
C     CALCULATING NEW COSTS
C
      DO 210 I=1,NM,1
      U(I+NM)=U(I)
      Z(I)=0.0
      DO 220 K=2,NG,1
      IF((1+N-K).GE.N)GOTO 220
      Z(I)=Z(I) + XQ(I,(1+N-K))*V(K)
220   CONTINUE
      Z(I+NM)=Z(I) + XQ(I+NM,N)*V(1)
      Z(I)=Z(I) + XQ(I,N)*V(1)
      U(I+NM)=U(I+NM) + (R(N)-Z(I+NM))*(R(N)-Z(I+NM))
      U(I)=U(I) + (R(N)-Z(I))*(R(N)-Z(I))
210   CONTINUE
C
C     SELECTING NM VECTORS
C
      J7=(NM-NL)/2
      DO 230 L=1,J7,1
      MINU=10E9
      MINV=10E9
      MINI=2*NM+1
      MINJ=2*NM+2
      J=N - ((NM-NL)/2) + L
      JS=NM*2
      DO 240 I=1,JS,1
      IF(XS(I,J).LT.1)GOTO 250
      IF(U(I).GT.MINU)GOTO 240
      MINU=U(I)
      MINI=I
      GOTO 240
250   IF(U(I).GT.MINV)GOTO 240
      MINV=U(I)
      MINJ=I
240   CONTINUE
      DO 260 JY=1,N,1
      YS(L,JY)=XS(MINI,JY)
      YS(L+(NM-NL)/2,JY)=XS(MINJ,JY)
      YQ(L,JY)=XQ(MINI,JY)
      YQ(L+(NM-NL)/2,JY)=XQ(MINJ,JY)
260   CONTINUE
```

```
      YU(L)=U(MINI)
      YU(L+(NM-NL)/2)=U(MINJ)
      U(MINI)=10E10
      U(MINJ)=10E10
230   CONTINUE
      IF(NL.LE.0)GOTO 270
      DO 280 II=1,NL,1
      MINU=10E9
      J9=NM*2
      DO 290 I=1,J9,1
      IF(U(I).GT.MINU)GOTO 290
      MINU=U(I)
      MINI=I
290   CONTINUE
      YU(II+(NM-NL))=U(MINI)
      U(MINI)=10E10
      DO 300 JY=1,N,1
      YS(II+(NM-NL),JY)=XS(MIN,JY)
      YQ(II+(NM-NL),JY)=XQ(MINI,JY)
300   CONTINUE
280   CONTINUE
C
C     DETECT XS AS S
C
270   MINU=10E9
      DO 310 I=1,NM,1
      IF(YU(I).GT.MINU)GOTO 310
      MINU=YU(I)
      MINI=I
310   CONTINUE
      ES1=YS(MINI,N-128+1)
      ES2=YS(MINI,N-96+1)
      ES3=YS(MINI,N-64+1)
      ES4=YS(MINI,N-48+1)
      ES5=YS(MINI,N-32+1)
      ES6=YS(MINI,N-16+1)
C
C     TRANSFER Y BACK TO X
C
      DO 320 I=1,NM,1
      U(I)=YU(I)
      DO 330 J=1,N,1
      XS(I,J)=YS(I,J)
      XQ(I,J)=YQ(I,J)
330   CONTINUE
320   CONTINUE
C
C     CALCULATE ERROR RATE
C
      IF(IA.LT.N)GOTO 888
```

```
      EQ=EQ + Q(1)*Q(1)
      IF(IC.LT.50)GOTO 340
      IF(ID.LT.20)GOTO 350
      WRITE(2,7400)(NER1(I),I=1,20,1)
      WRITE(2,7400)(NER2(I),I=1,20,1)
      WRITE(2,7400)(NER3(I),I=1,20,1)
      WRITE(2,7400)(NER4(I),I=1,20,1)
      WRITE(2,7400)(NER5(I),I=1,20,1)
      WRITE(2,7400)(NER6(I),I=1,20,1)
      WRITE(2,8600)
      DO 360 I=1,20,1
      NER1(I)=0
      NER2(I)=0
      NER3(I)=0
      NER4(I)=0
      NER5(I)=0
      NER6(I)=0
360   CONTINUE
      ID=1
      GOTO 370
350   ID=ID+1
370   IC=1
      GOTO 380
340   IC=IC+1
380   IF(ES1*5(N-128+1))400,400,410
400   ERROR1=ERROR1+1
      NER1(ID)=NER1(ID)+1
410   IF(ES2*5(N-96+1))420,420,430
420   ERROR2=ERROR2+1
      NER2(ID)=NER2(ID)+1
430   IF(ES3*5(N-64+1))440,440,450
440   ERROR3=ERROR3+1
      NER3(ID)=NER3(ID)+1
450   IF(ES4*5(N-48+1))460,460,470
460   ERROR4=ERROR4+1
      NER4(ID)=NER4(ID)+1
470   IF(ES4*5(N-32+1))480,480,490
480   ERROR5=ERROR5+1
      NER5(ID)=NER5(ID)+1
490   IF(ES6*5(N-16+1))500,500,888
500   ERROR6=ERROR6+1
      NER6(ID)=NER6(ID)+1
888   CONTINUE
C
C     CALCULATE SNR IN DB
C
      ENI=NI*1.0
      EQ=EQ/ENI
      IF(SD.LE.0.0)GOTO 510
      SNR=10.0*(ALOG10(EQ/SD/SD))
```

```
      GOTO 520
510   WRITE(2,8200)
520   WRITE(2,9000)
      WRITE(2,7000)NG,NM,NL,N,NI,NB,NH,NK
      WRITE(2,7100)SD,NON
      WRITE(2,7200)(Y(I),I=1,NG,1)
      DO 530 I=1,NH,1
      WRITE(2,7300)(G(I,J),J=1,NK,1)
530   CONTINUE
      WRITE(2,9000)
      WRITE(2,9100)NI
      WRITE(2,9201)ERROR1
      WRITE(2,9202)ERROR2
      WRITE(2,9203)ERROR3
      WRITE(2,9204)ERROR4
      WRITE(2,9205)ERROR5
      WRITE(2,9206)ERROR6
      WRITE(2,9300)EQ
      WRITE(2,9400)SD
      WRITE(2,9500)SNR
      IF(IB.GE.NB)GOTO 540
      EQ=0.0
      ERROR1=0
      ERROR2=0
      ERROR3=0
      ERROR4=0
      ERROR5=0
      ERROR6=0
      READ(1,*)SD
      IB=IB+1
      WRITE(2,8300)
      DO 550 I=1,NM,1
      WRITE(2,8400)(XS(I,J),J=1,N,1)
      WRITE(2,8500)YU(I)
550   CONTINUE
      GOTO 999
540   STOP
      END
      SUBROUTINE ENCO(NA,NB,DN,NN,NNH,NNK)
      DIMENSION NA(130),NB(2,10),NC(2)
      DO 5000 ISE=1,NN,1
      IF(NA(ISE).EQ.1)GOTO 5000
      NA(ISE)=0
5000  CONTINUE
      DO 5100 ISE=1,NNH,1
      NC(ISE)=0
      DO 5200 JSE=1,NNK,1
      NC(ISE)=NC(ISE) + NA(NN-JSE+1)*NB(ISE,JSE)
      IF(NC(ISE).LT.2)GOTO 5200
      NC(ISE)=NC(ISE)-2
```

```
5200   CONTINUE
5100   CONTINUE
       IF(NC(1).EQ.0.AND.NC(2).EQ.0)GOTO 5300
       IF(NC(1).EQ.0.AND.NC(2).EQ.1)GOTO 5400
       IF(NC(1).EQ.1.AND.NC(2).EQ.0)GOTO 5500
       DN=1.0
       GOTO 5600
5300   DN=-3.0
       GOTO 5600
5400   DN=-1.0
       GOTO 5600
5500   DN=+3.0
5600   RETURN
       END
```