TECHNIQUES OF DETECTION, ESTIMATION AND CODING FOR FADING CHANNELS

by

Sankissa G. Jayasinghe

A doctoral thesis submitted in partial fulfilment of the requirements
for the award of
Doctor of Philosophy of the Loughborough University of Technology.

Supervisor : *Professor A. P. Clark*
Department of Electronic and Electrical Engineering
Loughborough University of Technology.

Everything should be made as simple as possible, but not simpler.

- Albert Einstein

To my wife, Sandra, and my mother, Swarna ;

A part of you lives in here

ABSTRACT

The thesis describes techniques of detection, coding and estimation, for use in high speed serial modems operating over fading channels such as HF radio and land mobile radio links. The performance of the various systems that employ the above techniques are obtained via computer simulation tests.

A review of the characteristics of HF radio channels is first presented, leading to the development of an appropriate channel model which imposes Rayleigh fading on the transmitted signal. Detection processes for a 4.8 kbit/s HF radio modem are then discussed, the emphasis, here, being on variants of the maximum likelihood detector that is implemented by the Viterbi algorithm. The performance of these detectors are compared with that of a nonlinear equalizer operating under the same conditions, and the detector which offers the best compromise between performance and complexity is chosen for further tests.

Forward error correction, in the form of trellis coded modulation, is next introduced. An appropriate 8-PSK coded modulation scheme is discussed, and its operation over the above mentioned HF radio modem is evaluated. Performance comparisons are made of the coded and uncoded systems.

Channel estimation techniques for fast fading channels akin to cellular land mobile radio links, are next discussed. A suitable model for a fast fading channel is developed, and some novel estimators are tested over this channel. Computer simulation tests are also used to study the feasibility of the simultaneous transmission of two 4-level QAM signals occupying the same frequency band, when each of these signals are transmitted at 24 kbit/s over two independently fading channels, to a single receiver. A novel combined detector/estimator is developed for this purpose.

Finally, the performance of the complete 4.8 kbit/s HF radio modem is obtained, when all the functions of detection, estimation and prefiltering are present, where the prefilter and associated processor use a recently developed technique for the adjustment of its tap gains and for the estimation of the minimum phase sampled impulse response.

ACKNOWLEDGEMENTS

My supervisor, Professor A. P. Clark, was the architect of this research project. Always at hand for advice, conversation and critical evaluation of theories and results, he ensured that successes were many, and blind alleys few. The welfare of the student was also one of his prime concerns and I was provided with an environment conducive to the progression of the research. For all this and much more, I am deeply grateful.

The financial support of the Ministry of Defence (GCHQ, Cheltenham, U.K.), and the Department of Electronic and Electrical Engineering at Loughborough University of Technology, is gratefully acknowledged.

Many friends and colleagues bore varied influences during the course of the project, some directly pertaining to the research and some otherwise. I thank all for helping me broaden my vision. Thanks are also due to the staff at the Computer Centre at Loughborough University of Technology, in general, and Geoff Harris, Rob Kirkwood and Dave Thomas, in particular, for their willingness to help and their understanding in the allocation of computer resources.

On a personal basis, I would primarily like to thank my mother, Swarna, for giving me, initially the foundation, and then the opportunity, to further my education. I hope her faith in me has been well justified. My thanks also extend to my stepfather, Mohan, for his selflessness and tolerance, especially during the early years, and to my parents-in-law, Quintus and Ruth, not only for ensuring that a students life need not necessarily be bereft of material excesses, but also for their unique understanding in all matters of consequence. To my father, Ratnakara, thank you for the extra dimension.

My wife, Sandra, proof-read the manuscript and made useful comments on the presentation. If those mentioned above gave me the support, it is she who gave me the reason. This work is as much her design as it is mine.

S.G.J

TABLE OF CONTENTS

CHAPTER 1 : INTRODUCTION 1

1.1 BASIC DEFINITIONS 1
1.2 BACKGROUND 2
1.3 OUTLINE OF THESIS 5
REFERENCES 7

CHAPTER 2 : THE HF RADIO CHANNEL 10

2.1 INTRODUCTION 10
2.2 THE IONOSPHERE 10
2.3 IONOSPHERIC PROPAGATION 13
2.4 SIGNAL IMPAIRMENTS OCCURING OVER HF RADIO CHANNELS 15
2.4.1 Multipath propagation 15
2.4.2 Fading 16
2.4.3 Frequency selectivity of channel 18
2.4.4 Doppler shifts 18
2.4.5 Noise 19
2.5 HF CHANNEL CLASSIFICATION 19
2.6 MODEL OF CHANNEL 20
2.6.1 Outline of model 21
2.6.2 Generation of the Q_i (t) 22
2.7 VERIFICATION OF CHANNEL MODEL 24
REFERENCES 28
TABLES AND FIGURES 30

CHAPTER 3 : DATA TRANSMISSION USING QUADRATURE AMPLITUDE MODULATION 47

3.1 INTRODUCTION 47
3.2 THE QAM SYSTEM MODEL 47
3.3 QAM SYSTEM MODEL WHEN USED OVER HF RADIO LINKS 56
REFERENCES 68

FIGURES 69

CHAPTER 4 : NEAR MAXIMUM LIKELIHOOD DETECTORS FOR A 4.8 KBIT/S HF RADIO MODEM 77

4.1 INTRODUCTION 77
4.2 THE VITERBI ALGORITHM 78
4.3 MODIFICATIONS AND DERIVATIVES OF THE VITERBI DETECTOR 84
4.4 MODEL OF THE DATA TRANSMISSION SYSTEM USED IN THE SIMULATION 88
4.5 THE DETECTORS 96
4.5.1 System A 96
4.5.2 System B 98
4.5.3 System C 101
4.6 COMPUTER SIMULATION TESTS AND RESULTS 102
REFERENCES 106
TABLES AND FIGURES 109

CHAPTER 5 : TRELLIS CODED MODULATION FOR A 4.8 KBIT/S HF RADIO MODEM 140

5.1 INTRODUCTION 140
5.2 TRELLIS CODED MODULATION 141
5.2.1 General description 141
5.2.2 Rotationally invariant codes 149
5.3 MODEL OF DATA TRANSMISSION SYSTEM USED IN THE SIMULATION 154
5.4 THE DECODERS 159
5.4.1 Viterbi algorithm decoders 159
5.4.2 System D 160
5.4.3 System E 163
5.5 COMPUTER SIMULATION TESTS AND RESULTS 163
REFERENCES 169
TABLES AND FIGURES 173

CHAPTER 6 : CHANNEL ESTIMATION FOR FAST FADING CHANNELS 187

6.1 INTRODUCTION 187
6.2 BASIC ASSUMPTIONS 188
6.3 MODEL OF SYSTEM 189
6.4 THE ESTIMATORS 191
6.4.1 The gradient estimator 192
6.4.2 The modified gradient estimator 193
6.4.3 The unbiased estimator 195
6.5 THE DETECTORS 200
6.5.1 The simple detector 200
6.5.2 The combined detector 200
6.6 COMPUTER SIMULATION TESTS AND RESULTS 204
REFERENCES 211
TABLES AND FIGURES 213

CHAPTER 7 : COMBINED DETECTION AND ESTIMATION FOR A 4.8 KBIT/S HF RADIO MODEM 229

7.1 INTRODUCTION 229
7.2 MODEL OF DATA TRANSMISSION SYSTEM 230
7.3 THE DETECTOR 234
7.4 THE ESTIMATOR 235
7.5 THE ADAPTIVE FILTER 237
7.6 COMPUTER SIMULATION TESTS AND RESULTS 238
REFERENCES 243
FIGURES 244

CHAPTER 8 : COMMENTS 248

8.1 SUMMARY OF PROJECT 248
8.2 CONCLUSIONS 249

APPENDICES 251

A : DERIVATION OF 5-POLE BESSEL FILTER 252
REFERENCES 256
TABLES AND FIGURES 257
B : DIFFERENTIAL ENCODING AND DECODING 259
TABLES AND FIGURES 261
C : SIGNAL-TO-NOISE RATIO CALCULATIONS 264
REFERENCES 271
D : UNBIASED ESTIMATES 272
D.1 Unbiased estimate of a fast fading channel 272
D.2 Unbiased estimate of a very fast fading channel 275
REFERENCES 279
E : THE EQUIVALENCE OF THE TESTED ESTIMATORS
WHEN ONE SIGNAL IS RECEIVED 280
F : ADAPTIVE ADJUSTMENT OF PREFILTER 282
REFERENCES 290
FIGURES 291
G : SIMULATION OF 3-SKYWAVE HF RADIO CHANNEL
INCORPORATING IDEAL MINIMUM PHASING 292
H : SIMULATION OF TRELLIS CODED MODULATION
SYSTEM D16 300
I : SIMULATION OF UNBIASED ESTIMATOR WITH DEGREE-1
PREDICTION AND PERFECT DETECTION 308
J : SIMULATION OF COMBINED DETECTOR AND MODIFIED
GRADIENT ESTIMATOR FOR FAST FADING CHANNELS 316
K : SIMULATION OF 4.8 KBIT/S HF RADIO MODEM
AS GIVEN IN CHAPTER 7 328

TECHNIQUES OF DETECTION, ESTIMATION

AND CODING

FOR FADING CHANNELS

CHAPTER 1

INTRODUCTION

1.1 BASIC DEFINITIONS

When historians of the future chronicle the events of the current millennium, they may probably look back to the 19th century as the era which sowed the seeds of the 'communications revolution'. The invention of the telegraph - Samuel Morse (1837), the telephone - Alexander Graham Bell (1874) and the establishment of the existence of radio waves - Heinrich Hertz (1887), are three events that would amply justify their thinking. Confirmation of the existence of radio waves, in particular, has given rise to innumerable possibilities for communication over very long distances. These possibilities were further confirmed in 1901, when Guglialmo Marconi successfully received at Newfoundland in Canada, a radio signal that had been transmitted from Cornwall in England.

Since then, radio channels have been used with increasing regularity for the transmission of information. A perennial problem faced by radio communication systems designers however, has been the fact that radio channels introduce adverse and time varying effects on the information bearing signal. These problems are never more amplified than when a phenomenon known as signal fading occurs. When a signal is transmitted over a radio channel, it may reach its destination via several paths with differing lengths and thus may be received as a sequence of signals delayed in time. This phenomenon is known as multipath propagation [1]. As a consequence of multipath propagation, the time delayed signals at the receiver may interfere with each other in either a constructive or destructive manner. Constructive interference is accompanied by a corresponding gain in the resultant signal level, while destructive interference is accompanied by a corresponding loss. This variation of the signal at the receiver is known as signal fading, and radio channels exhibiting this phenomenon are sometimes known as fading channels.

Fading channels may themselves be broadly categorised as slow fading channels and fast fading channels. An example of the former is the high frequency (HF) radio channel, which normally occupies a band of frequencies between 3 and 30 MHz. (generally referred to as the HF radio spectrum). A typical fast fading channel is a cellular mobile radio channel which occupies a frequency band close to 900 MHz [2].

A device that performs the various functions needed for the transmission and

reception of data is called a modem. When the data is processed digitally, the modem itself is called a digital modem. Signal detection, error control coding and channel estimation are three of the many possible functions that may be performed in a digital modem. This thesis is concerned with the development of techniques of detection, coding, and estimation for digital modems operating over HF radio channels and fast fading channels.

1.2 BACKGROUND

The conventional HF radio modems are parallel modems, where transmission takes place over a number of subchannels within a 3 kHz. band, at a fairly low (eg. 75 baud) rate [3]. This tends to nullify to a great extent, the effects of multipath propagation. However, the performance, in general, of parallel HF radio modems have been shown to be inferior to that of their counterpart, the serial modem [4]. Indeed, with the advent of faster processors and new technology in the form of optimum detection schemes and faster adaptation algorithms, the emphasis is shifting inexorably towards the design of serial modems for HF radio communications.

In high speed, serial data transmission over a 3 kHz. band in the HF radio spectrum, one of the major problems facing the modem designer is that of eliminating the intersymbol interference (which is caused by multipath propagation) in the received signal. Of course, the fact that the channel characteristics vary with time, merely serves to compound this problem. Traditionally, such problems have been overcome by equalizing the received signal before passing it on to a detector [5]. The time varying nature of the HF channel is accommodated by making the equalizer adaptive, that is, by using various adaptation algorithms to adjust the tap gains of the equalizer at the appropriate time instants. However, the optimum detection process for a sequence of data symbols transmitted over a non-ideal, bandlimited channel which introduces intersymbol interference and additive white Gaussian noise, and where the transmitted symbols are equally likely to have any one of their possible values, is maximum likelihood detection. It has been shown that a maximum likelihood detector can be efficiently implemented via a recursive algorithm known as the Viterbi algorithm, and in these cases, it is alternately known as a Viterbi detector [5,6].

Although an implementable design, the Viterbi detector suffers from the problems of excessive storage requirements and high computational complexity, whenever a multilevel signal is used and/or the sampled impulse response of the channel has more than a few components. A class of detectors that reduce this problem to a certain extent, are known as near maximum likelihood detectors. Near maximum likeiihood detectors are

derivatives of the Viterbi detector and although suboptimal, it has been shown that [8], when operating on a binary or quaternary signal, they are not significantly inferior to Viterbi detectors, in terms of their tolerance to additive white Gaussian noise. This thesis examines the performance of various near maximum likelihood detectors, operating in a 4.8 kbit/s HF radio modem employing 4-level quadrature amplitude modulation.

The performance of a modem may be improved by introducing redundancy into the transmitted message and exploiting this redundancy at the receiver in a process known as error control coding [9]. When the coding scheme has the ability to correct as well as detect errors, it is known as error correction coding, and when no feedback path to the transmitter is used for this purpose, it is known as forward error correction coding (FEC).

Forward error correction schemes are generally accompanied by a corresponding increase in the required transmission bandwidth, or a corresponding decrease in the transmitted symbol rate. Thus, power efficiency may be achieved, but only at the expense of bandwidth efficiency. However, in 1976, Ungerboeck proposed a forward error correction scheme that does not change either the bandwidth or the symbol rate [10]. This type of coding is now widely known as trellis coded modulation (TCM) [11], a name arising from the fact that the convolutional encoder and modulator are now treated as one entity.

Trellis coded modulation systems gain an advantage over the corresponding uncoded modulation systems, when operating over additive white Gaussian noise channels. Indeed, they have been proved to be very successful in both satellite and telephone modems, [12,13] and their performance over the more volatile, fading, HF radio channels, is thus of tremendous interest. This thesis examines the performance of a suitable TCM system, when used in a 4.8 kbit/s HF radio modem.

A near maximum likelihood detector requires knowledge (as indeed does a Viterbi detector), not only of the possible transmitted data symbol values, but also of the sampled impulse response of the channel. The possible transmitted data symbol values are, of course, determined by the particular modulation method that is used. The sampled impulse response of radio channels, however, are continuously varying quantities and as such, even if determined at the beginning of the transmission, cannot be assumed to be constant over the duration of the transmission. The receiver, therefore, must somehow gain knowledge of the sampled impulse response of the channel at every appropriate sampling instant.

One approach toward solving this problem has been to employ a prefilter before

the detector, where the function of the prefilter is to generate a known, shortened impulse response, in the course of adapting itself to the time varying channel according to some error (least mean square - LMS, recursive least square - RLS) criterion [14,15]. However, this approach has its problems. Firstly, in trying to shorten the impulse response, the prefilter may cause some amplitude equalization of the channel, thereby enhancing the noise at its output. Secondly (and more importantly), the convergence of the tap coefficients, regardless of the particular algorithm (LMS, RLS, etc.) used for adaptation, is likely to be too slow for operation in a 4.8 kbit/s HF radio modem.

The alternative approach is to track the sampled impulse response of the channel with the aid of a channel estimator [16], which can be made to operate well, and without any complex equipment. A channel estimator does not perform any linear modification of the channel itself, and as such, fast convergence of its taps may be achieved with relatively simple algorithms. A channel estimator that is very simple to implement is the gradient estimator [17] which uses the least mean square (LMS) algorithm [18] to give an estimate of the sampled impulse response of the channel. However, the gradient estimator assumes that the channel is time invariant, or that it varies only very slowly with time. As such, it needs to be modified for effective use over fading channels. A modification of the gradient estimator that has been developed for HF radio links involves the prediction of future values of the sampled impulse response of the channel [19]. This thesis examines the performance of the above mentioned modified gradient estimator when operating over fast fading channels, and develops a novel estimator for use over slow, fast and very fast fading channels. The thesis also develops a novel combined detector-estimator for operation over fast fading channels.

Another requirement for near optimum performance in a near maximum likelihood detector, is that the magnitude of the first few components of the sampled impulse response of the channel should be large, relative to the other components. If the first component is the largest, then the sampled impulse response is referred to as a minimum phase response [5]. If the response is not minimum phase (or near minimum phase), then the near maximum likelihood detector must be suitably modified to take this into account, or some sort of processing is required ahead of the detector, to make the sampled impulse response of the channel minimum phase. Tests have shown that the former method leads to very complex systems which appear not to be able to handle severe phase distortion [20], and hence, the near maximum likelihood detectors in this thesis are designed on the basis that the sampled impulse response of the channel is minimum (or near minimum) phase.

The sampled impulse response of the channel may be made minimum phase

by means of a prefilter located at the input to the detector [5] . An adaptive linear feedforward transversal filter that performs the function of this prefilter, has been recently developed for telephone modems [21]. The sole input used by this filter is an estimate of the sampled impulse response of the channel, which, of course, is provided by the channel estimator. The final work in the thesis describes the operation and performance of the previously mentioned 4.8 kbit/s HF radio modem, when the detector operates in conjunction with an estimator, and the adaptive linear feedforward transversal filter.

1.3 OUTLINE OF THESIS

The thesis begins, in Chapter 2, with a description of the HF radio channel, and the possible signal impairments that could occur when a data signal is transmitted over such a channel. A model of the HF radio channel is developed and verified here, for use in the tests carried out in the thesis.

Chapter 3 gives the derivation for the model of a QAM data transmission system, and extends this theory to obtain a model of a QAM data transmission system that can be used over HF radio links.

Chapter 4 describes near maximum likelihood detection in some detail. The operation of the near maximum likelihood detectors used in the thesis is described, and the performance of these detectors, when operating in a 4.8 kbit/s HF radio modem, is assessed. Perfect channel estimation is assumed here, and the channel is made a minimum phase by some 'ideal' method. As such, the tests do not assume the use of a prefilter or a channel estimator.

The design of trellis codes for TCM is described in Chapter 5. A code suitable for transmission over HF radio links is obtained, along with two corresponding near maximum likelihood decoding schemes. This code is used to compare the performance of trellis coded modulation against that of uncoded modulation, when both the coded and uncoded systems are used in a 4.8 kbit/s HF radio modem.

Chapter 6 considers the problem of channel estimation for fast fading channels. The theory is given for the modified gradient estimator and a novel estimator known here as the 'unbiased estimator', where the latter could be adjusted to be used over slow, fast or very fast fading channels. A novel, combined detector-estimator is also developed in this chapter. The performance of the estimators is first compared under the assumption of perfect detection, after which detection is introduced to obtain and assess the performance

of the complete system. Two cases are considered here, these being the reception of one, and two signals over the given frequency band. In the latter case, the two signals are assumed to be fading independently, and no error correction or diversity is used to improve the performance of the system.

The final work in the research project is contained in Chapter 7 of the thesis, and concerns the operation and performance of the 4.8 kbit/s HF radio modem, when all the functions of detection, prefiltering and estimation are assumed to be present.

Chapter 8 gives a summary of the project and presents the main conclusions drawn from the research.

The thesis contains several appendices on descriptions of various algorithms, proofs and techniques, all of which, in the author's view, would have impeded the flow of the discussion, had they been included in the main text. The appendices also include source code listings of the main computer programs that were developed by the author, during the course of the research. All programs are in Fortran 77, and conform to the ANSI standards.

References - Chapter 1

1. **Stein, S.,** 'Fading channel issues in systems engineering', IEEE Journal on selected areas in Communications, vol. SAC-5, no. 2, pp. 68 - 89, February 1987.

2. **Arredondo, G.A., Feggeler, J.C. and Smith, J.I.,** 'Advanced mobile phone service: Voice and data transmission', Bell Systems Technical Journal, vol. 58, pp. 97 - 122, January 1979.

3. **Mosier, R. and Clabaugh, R.G.,** 'Kineplex: a bandwidth efficient binary data transmission system, AIEE Trans. on Commun. Electron., no. 76, pp. 723 - 728, 1958.

4. **Clark, A.P. and McVerry, F.,** 'Performance of 2400 bit/s serial and parallel modems over an HF channel simulator', IERE Conf. Proc. no. 49, pp. 167 - 179, April 1981.

5. **Clark, A.P.,** 'Equalizers for digital modems', Pentech Press, 1985.

6. **Forney, G.D.,** 'The Viterbi algorithm', IEEE Proceedings, vol. 61, no. 3, March 1973.

7. **Clark, A.P., Abdullah, S.N., Jayasinghe, S.G. and Sun, K.H.,** 'Psuedobinary and psuedoquaternary detection processes for linearly distorted multilevel QAM signals', IEEE Trans. on Communications, vol .COM-33, no .7, pp. 639 - 645, July 1985.

8. **Clark, A.P., Harvey, J. D. and Driscoll, J.P.,** 'Near maximum likelihood detection processes for distorted digital signals', Radio and Electronic Engineer, no. 48, pp. 30 - 309, June 1978.

9. **Lin, S and Costello, D.J.,** 'Error control coding: Fundamentals and applications', Prentice Hall, 1983.

10. **Ungerboeck, G.,** 'Channel coding with multilevel/phase signals', IEE Int. Symposium on Information Theory, Ronneby, Sweden, June 1976.

11. **Ungerboeck, G.,** 'Trellis coded modulation with redundant signal sets; Part I: Introduction, Part II: state of the art', IEEE Communications magazine, vol. 25, pp. 5 - 21, February 1987.

12. **Lebowitz, S.H. and Rhodes, S.A.,** ' Performance of coded 8-PSK signalling for sattelite communications', IEEE Int. Conf .on Communications (ICC '81) record, pp. 47.4.1- 47.4.8, Denver, Colorado, USA, June 1981.

13. **Brownlie, J.D. and Cusack, E.L.,** 'Duplex transmission at 4800 and 9600 bit/s on the general switched telephone network and the use of channel coding with a partitioned signal constellation', British Telecom Research Labs Report, 1984.

14. **Falconer, D.D. and Magee Jr., F.R.,** 'Adaptive channel memory truncation for maximum likelihood sequence estimation', Bell Systems Technical Journal, vol. 52, pp. 1541 - 1562, November 1973.

15. **Messerschmitt, D.G.,** 'Design of a finite impulse response for the Viterbi algorithm and decision feedback equalizer', IEEE Int. Conf. on Communications (ICC '74) record, pp. 37D.1 - 37D.5, June 1974.

16. **Clark, A.P., Kwong, C.P. and McVerry, F.,** 'Estimation of the sampled impulse response of a channel', Signal Processing, vol. 2, no. 1, pp. 39 - 53, January 1980.

17. **Magee, F.R. and Proakis, J.G.,** 'Adaptive maximum likelihood sequence estimation for digital signals in the presence of intersymbol interference', IEEE Trans. on Information Theory, vol. IT-19, pp. 120 - 124, January 1973.

18. **Widrow, B and Hoff, M.E. Jr.,** 'Adaptive switching circuits', IRE WESCON Conv. Rec., Pt. 4, pp. 563 - 587, 1960.

19. **Clark, A.P. and McVerry, F.,** 'Channel estimation for an HF radio link', IEE Proceedings, no. 128, Pt. F, pp. 33 - 42, Febraury 1981.

20. **Clark, A.P. and Najdi, H.Y.,** 'Detection process of a 9600 bit/s serial modem for HF radio links', IEE Proc., Part F, vol. 130, no. 5, pp. 368 - 376, August 1983.

21. **Clark, A.P. and Hau, S.F.,** 'Adaptive adjustment of receiver for distorted digital signals', IEE Proc., Part F, no. 5, pp. 536 - 536, August 1984.

CHAPTER 2

THE HF RADIO CHANNEL

2.1 INTRODUCTION

The spectrum between 3 and 30 MHz. is termed the high frequency (HF) radio spectrum. A voiceband, HF radio channel normally occupies about 3.5 kHz within this spectrum. In such channels, radio waves are propagated primarily due to reflection from the ionosphere. Thus, in order to model an HF radio channel, it is first necessary to have some sort of understanding of the ionosphere and its related phenomena.

Toward this end, the chapter begins with a brief description, in Section 2.2, of the formation and composition of the ionosphere. Section 2.3 explains the mechanics of radio wave propagation through the ionosphere. This explanation is somewhat simplified, (for example, it assumes a flat earth!), but nevertheless, should serve its purpose in introducing the reader to the nature and characteristics of the ionosphere.

Section 2.4 discusses the most common and significant signal impairments which occur on HF radio channels. These are discussed in some detail since any HF channel model should be capable of simulating such impairments and as such, a proper understanding of this is deemed necessary. Section 2.5 classifies the HF radio channels used in the thesis and in Section 2.6, the channel model is developed. The chapter ends in Section 2.7, with the verification of the simulated channel model.

2.2 THE IONOSPHERE

The ionosphere is a region of ionized air molecules, lying upwards of about 50 km. above the surface of the Earth. Free electrons exist here in sufficient abundance to influence the propagation of radio waves [1-4].

Ionization of the upper atmosphere is primarily caused by electromagnetic radiation from the Sun. The Earths atmosphere consists of molecules of nitrogen, oxygen and other rarer gases. When a photon, arising due to solar radiation, collides with a molecule, it can force an electron out of the molecule provided it has sufficient energy. Thus, the upper regions of the ionosphere would show a higher level of ionization than the lower regions due to the increased absorption, and hence decreased energy, of the solar

radiation as it penetrates the atmosphere. Ionization is also caused by charged particles of solar origin entering the Earths atmosphere, meteors travelling through this atmosphere and by cosmic rays. Indeed, it is generally accepted that cosmic rays are the principal source of ionization in the lower regions of the ionosphere where the solar radiation is very weak. Ionization of the Earths atmosphere is counterbalanced by the recombination of electrons and positive ions and by the attachment of electrons to neutral gas atoms and molecules. Thus, the level of ionization would depend not only on the intensity of the solar radiation but also on the frequency of collisions between electrons and other particles.

From this rather simplistic view of the formation of the ionosphere, it would appear that the electron density due to ionization should increase gradually with altitude. However, the situation in practice is much more complex, primarily due to the variable nature of the constitution of the Earths atmosphere. Studies of the ionosphere have shown[1-4] that the level of ionization, and hence the electron density, does increase with height up to about 400km but not in any simple, uniform manner. Rather, the ionization is distributed in layers, where each layer displays a higher level of ionization than the layer immediately below it. This is accompanied by a decrease, with altitude, in the number of collisions between electrons and other particles. As a result, the peak electron densities in the upper layers are greater than those in the lower layers. Beyond an altitude of about 450 km. however, the electron density actually decreases with height, aided, no doubt, by a relative increase in the collision frequency, and other complex phenomena [5-7].

The ionosphere has classically been divided into 3 such layers, namely, D, E, and F, with subdivisions existing under certain conditions, notably, the F_1 and F_2 layers. The salient parameters of these layers are their virtual heights, electron densities and critical frequencies, where the latter is defined as the highest frequency of a vertically incident ray that can be reflected by the layer. However, due to the volatile nature of the ionosphere, the values of these parameters change both temporally and spatially, and as such, any value given should be taken as typical and not absolute [5] (see Table 2.2.1). The temporal variations occur daily, seasonally and according to the 11-year solar cycle, while the spatial variations occur according to the geomagnetic latitude. Fig. 2.2.1 shows a typical electron density profile at mid latitudes during summer noon, at solar maximum conditions [7], and Fig. 2.2.2 shows the ionospheric layers as a function of height above the surface of the Earth [8].

The D-layer has a high molecular concentration of the order of 10^{20} molecules/m^3 and, as seen in Table 2.2.1, a relatively low electron density. This results in a very high number of collisions between electrons and molecules, making this region essentially an attenuation band for HF radio waves. However, VLF, LF and MF radio

waves can be reflected from this layer. The maximum electron density in this region is observed at local noon during summer. The D-layer is a daytime phenomenon and is virtually non existent at night (see Fig. 2.2.2).

The E-layer has a lower ionization absorption level, and hence, a higher overall electron density than the D-layer, and is a good reflector of HF radio waves. It can support propagation upto distances of about 2000 km. The maximum electron density occurs at local noon during summer and in general, peak electron densities occur at about 110 km. above the surface of the Earth. Although the E-layer does exist at night, the ionization level is generally small and is more conducive to the propagation of MF, rather than HF, radio waves.

Embedded within the E-layer there exists, from time to time, patches of abnormally high levels of ionization. These are known as sporadic E or E_s-layers [1-4]. Their properties and frequency of occurrence differ significantly with geomagnetic latitude. An E_s-layer is capable of reflecting very high frequencies (> 10 MHz.), but is generally regarded as a nuisance for HF radio wave propagation due to its 'sporadic' occurrence and highly volatile and unreliable nature.

The portion of the ionosphere most suited for HF radio communication is the F-region. Due to the different variations observed in the lower and upper parts of this region, it has been further subdivided into the F_1 and F_2 layers. However, at night, these two layers merge to form a single F-layer which can support propagation upto distances of 4000 km. and is the only part of the ionosphere which could be used for reliable nighttime communication.

The F_1-layer behaves in a similar fashion to the E-layer, with the bulk of the ionization occurring due to solar radiation, and peak electron densities occurring just after local noon during summer. On the other hand, the behaviour of the F_2-layer is completely different to that of the other layers and its ionization does not bear a direct relationship to the intensity of the solar radiation. The uniqueness of the F_2-layer is manifested in the many anomalies it exhibits [7], such as the diurnal anomaly (maximum electron density occurring well after noon), and the seasonal anomaly (electron densities at noon in winter exceeding the corresponding values during summer). However, the highest ionization levels are found in the F_2-layer and as such, it plays a very important role in the propagation of HF radio waves through the ionosphere.

2.3 IONOSPHERIC PROPAGATION

An HF radio wave propagated through the ionosphere undergoes absorption, refraction and a phenomenon known as ray splitting. As explained previously, a high level of absorption occurs in the lower regions of the ionosphere, such as in the D-layer, due to the relative abundance of neutral gas molecules. However, absorption is also proportional to $1/f^2$ [3], where f is the frequency of the radio wave, and hence by increasing f, the attenuation due to absorption could be minimised.

Propagation of radio waves is possible due to progressive refraction at different heights in the ionosphere. The refractive index of the ionized medium is given by η, where[1],

$$\eta = \left\{ 1 - \frac{N e_e^2}{E_0 m_e \omega^2} \right\}^{1/2} \qquad (2.3.1)$$

and

N = electron density in electrons/m^3

e_e = charge of an electron = 1.6×10^{-19} C

E_0 = permittivity of free space = $10^{-9} / 36\pi$ F/m

M_e = mass of an electron = 9×10^{-31} kg

ω = angular frequency of radio wave.

Substituting the values of e_e, E_0 and m_e gives

$$\eta = \left(1 - \frac{81 N}{f^2} \right)^{1/2} \qquad (2.3.2)$$

where $f = \omega/2\pi$ = frequency of radio wave in MHz.

Since the electron density is a function of the height above the surface of the

Earth, it could be seen from eqn. 2.3.2, that the refractive index η, would change continuously with height, giving rise to the progressive refraction of radio waves. These radio waves are returned to Earth, as shown in Fig. 2.3.1, when N is large enough to reduce the value of η such that [6],

$$\eta = \sin\theta \qquad (2.3.3)$$

where θ is the angle of incidence of the radio wave with the ionized layer. For vertical incidence, $\theta = 0$ and hence, $\sin\theta = 0$. From eqns. 2.3.2 and 2.3.3,

$$\left(1 - \frac{81\,N}{f^2}\right)^{1/2} = 0$$

i.e.,

$$f = 9\sqrt{N} = f_0 \qquad (2.3.4)$$

where f_0 is, of course, the critical frequency as defined in Section 2.2 and is given in MHz. However, when a radio wave is obliquely incident on an ionospheric layer, as is necessary for HF radio communication, frequencies higher than f_0 are reflected according to the relationship

$$f = f_0 \sec\theta \qquad (2.3.5)$$

For a given angle of incidence, θ_1, there exists a maximum frequency which will be reflected by a particular layer. This frequency is known as the maximum usable frequency (MUF) of the layer and is given by

$$f\,(\mathrm{MUF}) = f_0 \sec\theta_1 \qquad (2.3.6)$$

Due to its dependence on f_0 , it is obvious that for a given angle of incidence, the MUF's of the higher layers are greater than those of the lower layers. Thus, whenever possible, the operating frequency of an HF radio link should be chosen such that it is close to the MUF of the layer via which propagation is desired. This is done in order to surpress unwanted reflections of the radio wave from the lower layers. Towards this end,

it is customary to employ an operating frequency called the optimum working frequency (FOT), which is taken to be approximately 85% of the MUF [1,3].

It should be noted that the above discussion on the progressive refraction of radio waves has been based on ray theory, assuming a flat Earth and neglecting the effect of the Earth's magnetic field. However, it is adequate for the purpose intended, which is to give the reader a broad indication of the manner in which HF radio waves are propagated through the ionosphere.

The presence of the Earths magnetic field gives rise to a phenomenon known as ray splitting, or magnetoionic splitting. When a radio wave enters the ionosphere, it interacts with the Earths magnetic field and is split into two differently polarised waves called the ordinary and extraordinary rays. These waves are reflected by a different electron density and as such, at slightly different heights from the ionosphere.

2.4 SIGNAL IMPAIRMENTS OCCURRING ON HF RADIO CHANNELS

When a data signal is transmitted over an HF radio link it is subjected to various types of time varying distortion and is corrupted by various forms of noise. The parameters that characterize an HF radio channel are based on these types of distortion and, thus, it is essential to understand them in order to appreciate the different classifications of such channels and indeed, to develop a channel model.

2.4.1. Multipath Propagation

As explained in Section 2.3, a radio wave could reach its destination via many propagation paths which are generally known as different modes of propagation. This could be due to reflection from two or more ionospheric layers, reflection from a single layer but with two or more hops, reception of two differently polarized waves (magnetoionic splitting), reception of two or more waves with different elevation angles connecting the same end points (high/low angle rays) or the existence of a groundwave in addition to the skywave [9]. These modes are illustrated in Fig. 2.4.1. Thus, the transmission of a very short pulse could result in the reception of a train of pulses, as shown in Fig. 2.4.2. Due to the difference in group delay in the various modes, there will be a time difference between the reception of the first and the last of these pulses. This time difference is known as the multipath delay spread or time spread of the HF radio channel.

Multipath delay spread is a major limiting factor in the operation of high speed HF radio modems. The delays that can be tolerated depend on the nature of the communication signal and its transmission rate. When the multipath spread becomes comparable to the signalling period, adjacent signal elements interfere with each other, causing a degradation in the error rate. This type of distortion is known as intersymbol interference and becomes serious when the multipath spread is 40% [10] or more, of the duration of a signal element. However, by judicious choice of the operating frequency, the number of propagation paths, and hence the multipath spread, can be reduced. This is because all paths in a given layer tend to coalesce when the operating frequency is close to the MUF for that layer.

Over medium distances, (800 - 2000 km.), multipath spreads of about 1 ms. could be encountered. More severe multipath spreads of upto 10 ms. occur over short distances and very long paths involving multiple hops. Furthermore, if the reflection points are in or near the auroral belt, it is not impossible to encounter spreads of nearly 10 ms [11]. However, most HF modems operate over long distances at mid and equatorial latitudes, and as such, are generally designed to accommodate multipath spreads of up to 3 ms.

2.4.2. Fading

Fading can broadly be divided into two types, long term and short term fading. Long term fading is the variation of the received signal strength, which is related to the daily, seasonal and annual variations in the structure of the channel and, in some cases, could even involve the 11-year sun spot cycle. However, it is the short term fading that is most important to the modem designer since this affects the received waveform structure within the time span of a transmitted message.

When an unmodulated sine wave carrier is transmitted over an HF radio channel, it could reach its destination via two or more paths due to multipath propagation. However, the signal received on each path (each skywave) is, in itself, made up of the sum of several slightly different paths, all adding randomly at the receiver. Thus, each skywave could be thought of as the sum of a number of vectors (phasors) having slightly different amplitudes and widely different phases [12]. (These large changes in the phase of the individual components constituting each skywave occur due to the small changes in their path lengths.) Furthermore, due to the short term variations in the structure of the ionosphere, these vectors are random and time variant. This results in the vectors at times adding destructively, and at other times adding constructively. When they add

destructively, the resultant received vector (or received skywave), becomes very small, or practically zero, and when they add constructively, it is observed to be very large. These variations in the received signal amplitude are generally known as signal fading, and they occur due to the time varying characteristics of the channel.

This type of fading is sometimes referred to as multiplicative fading or flat fading. The term flat fading arises due to the fact that all frequency components in the signal are affected in a similar manner. Furthermore, the envelope of the faded signal is observed to follow a Rayleigh distribution, with the phase being uniformly distributed (see Fig. 2.4.3.), giving rise to the term 'Rayleigh fading'. In some cases, other distributions have been found to fit the envelope of a faded signal, most notably, the Rician distribution (occurring when a specular component, such as a ground wave, is present in addition to the sky waves) and the Nakagami-n distribution [11]. However, over HF channels, the most widely observed type of fading is Rayleigh fading, and hence, the work in this thesis will be based on channels which exhibit such.

A single tone, when Rayleigh faded, gives rise to a spectrum of nonzero width as shown in Fig. 2.4.3. Thus, the channel introduces a 'spread' into the frequency of the transmitted signal. This is another parameter of an HF channel and is called the frequency (or Doppler) spread. Its magnitude, in particular, is closely related to the fading rate [13]. For single hop links over medium distances the frequency spread rarely exceeds 0.01 Hz.[11], this constituting very slow fading. More severe spreads, ranging upto 2 Hz., occur over long distances involving multiple hops. Although in some cases, spreads of even upto 20 Hz. have been observed to occur (especially in the polar regions), most HF modems are designed to combat fading that gives rise to frequency spreads of upto 2 Hz.

Adjacent channel interference, occurring due to these frequency spreads, could, at this stage, be thought of as a potential problem. However, most signal bandwidths of practical interest are much larger than the spreads introduced to their spectra and hence, interference between adjacent channels, if any, is negligible.

The fading rates associated with these frequency spreads are generally of the order of 10-50 fades per minute, while fade depths are usually less than 20 dB. It should be noted however, that fading rates could rise to more than 1 fade per second and on occasion, fade depths of over 30 dB have been encountered [9].

2.4.3. Frequency selectivity of channel

The coherence bandwidth of an HF radio channel is a measure of the frequency coherence of the channel, and is taken to be approximately equal to the reciprocal of the total multipath spread. (For a more formal definition of the coherence bandwidth, the reader is referred to Proakis, 1983). When the bandwidth of a signal transmitted over an HF radio channel lies within the coherence bandwidth of the channel, all parts of the signal spectrum are affected equally by the fading. This is flat fading, and the channel is said to be frequency nonselective. However, when the signal bandwidth exceeds the coherence bandwidth, the spectrum of the signal is subjected to different gains and shifts across the band, and the channel is now said to be frequency selective. Thus, two sinusoids with a frequency separation greater than the coherence bandwidth are affected differently by the channel.

An equivalent statement for frequency selectivity, (especially for simple symbol waveforms where the symbol duration is approximately equal to the reciprocal of the signal bandwidth), is that the multipath spread should be greater than the symbol duration. Note that this is also a condition, as explained in Section 2.4.1, for intersymbol interference and as such, signals transmitted over frequency selective channels are severely distorted.

A frequency selective channel could thus be viewed as one which gives rise to two or more skywaves at the receiver, where each has a different (and resolvable) multipath delay, and is affected differently by the channel. In HF radio transmission, as indeed in most time variant communication, the attenuation and phase shift associated with one skywave is generally uncorrelated to those associated with another skywave [12]. Thus, the fading on each skywave is uncorrelated and, under certain conditions [14], is considered to be independent.

When a signal transmitted over a frequency selective channel undergoes fading, the fading in itself is sometimes called frequency selective fading, in contrast with flat fading which is observed on frequency nonselective channels. Note however, that frequency selectivity is inherently dependent on the multipath spread, while flat fading occurs as a result of the time variations of the channel.

2.4.4. Doppler Shifts

Due to the continuous movement of the ionospheric layers, a signal transmitted over an HF channel undergoes a mean shift in frequency called the Doppler shift. In

general, Doppler shifts range from 0.1 Hz. to 1 Hz. and become quite small during nighttime. Under extreme conditions, (for example, during solar flares), values of upto 50 Hz. have been observed [9], but as regards HF modem design, the Doppler shifts which a modem need tolerate are taken to be their typical values. Note that the above concerns Doppler shifts, which occur due to moving platforms, and not Doppler spreads, which arise as a consequence of signal fading.

2.4.5 Noise

The main types of additive noise encountered over HF radio links are atmospheric noise and impulsive noise [15]. Impulsive noise is man-made interference and becomes important only near built-up areas. Atmospheric noise on the other hand, could be a major source of performance degradation in an HF radio modem. Occurring due to lightning discharges, atmospheric noise has a bandwidth extending to around 30 MHz.[15]. A bandlimited, received signal would thus contain noise components originating from several lightning discharges, each of which could be assumed to be a statistically independent source. Central limit theorem arguments lead to the assumption that this noise has a Gaussian probability density function, at least near the mean, which is zero. This assumption has held the HF radio modem designer in good stead, since it has been found out that a modem having a better tolerance to additive Gaussian noise would almost certainly have a better tolerance to atmospheric noise [15]. Thus, performance comparisons of modems are carried out on the basis of their tolerance to additive Gaussian noise, and the work in this thesis is no exception.

2.5 HF CHANNEL CLASSIFICATION

There is at present, no standard classification for HF radio channels, probably due to the number and variety of parameters that need to be considered (and indeed their relative importance) in such classifications. The CCIR, for example, have classified HF channels according to their multipath delays and frequency spreads as good, average, poor and flutter fading channels [14] (see Table 2.5.1.). These channel models are based on two skywaves where each skywave is subjected to independent Rayleigh fading, and has same frequency spread.

The channels used in this thesis are based on the CCIR model. Channel 1 is a three-skywave channel with a frequency spread of 2 Hz. on each skywave and relative delays of 1.1 ms. and 3.0 ms. between them. Channels 2 and 3 are two-skywave channels

with parameters as given in Table 2.5.2. Note that Channel 2 is similar to the CCIR 'flutter fading' channel (albeit having a more realistic value for the frequency spread), and Channel 3 is exactly the same as the CCIR 'poor' channel. All three channels could be said to be 'worst case' ones, and were thus chosen so that a detection or coding process exhibiting satisfactory performance over such a channel would probably be suitable for inclusion in a modem operating over a real HF radio channel.

2.6 MODEL OF CHANNEL

The performance of HF radio modems could be ascertained either by direct on-the-air measurements or by testing the modems over an HF fading channel simulator in a laboratory environment. Although it would seem, at first thought, that the former method is by far the more accurate (since testing is being carried out over 'real' channels), it is not necessarily so, and in any case, it suffers from quite a few disadvantages. For example, it is virtually impossible to compare the performance of different modems accurately since, due to the time varying nature of the HF radio medium, they cannot be tested over the same channel. It is also difficult to test modems for 'worst case' conditions since these could occur at random, making scheduled testing impossible. Even when tests are carried out regardless of these restrictions, it is sometimes difficult to pinpoint the exact reason for the poor performance of a system since this could be due to any one of a multitude of effects such as multipath spread, fading, Doppler shifts and atmospheric noise, to name a few. Consequently, the design cannot be improved if the relative importance of the aforementioned effects for a particular link are not known.

By employing an HF channel simulator, almost all of these constraints could be nullified. Laboratory simulators exist in hardware or software [8], the essential difference here being that, in a software simulator, the mechanics of the process are converted to changes in the instructions in a computer simulation program. These channel simulators have the advantages of repeatability, availability of any appropriate set of channel conditions as desired, flexibility in introducing distortion, and lower cost. However, care should be taken to ensure that the model on which the simulator is based is a valid representation of an HF radio channel. As such, the modem designer should be fully aware of the assumptions made in modelling the channel, and the conditions, if any, under which the model could be considered to be invalid.

2.6.1 Outline of model

The HF channel model used in this thesis is based on that proposed by Watterson, et. al, [16] and adopted by the CCIR [14]. This model has been proved to be a valid representation of the majority of HF radio channels. It is based on the assumption that the channel is narrowband (< 10 kHz.) and that sufficiently short transmission times (< 10 mins.) are considered. Both these assumptions do not, in any way, invalidate the work carried out in this thesis since the channels of interest here are bandlimited voiceband channels (≈ 3 kHz.), and transmission times are in the order of tens of seconds rather than minutes. It should be noted here, that it is not the intention of the author to either physically justify or experimentally validate the channel model since this would only involve the repetition of already established work [16]. Rather, the objective is to present the channel model in its given form, ensuring that the assumptions made are both reasonable and valid, for the research being carried out.

Figure 2.6.1 gives an illustration of the channel model. This is a stationary model (a valid assumption for HF channels, given the constraints mentioned above on bandwidth and transmission times) with a number of discrete modes representing the different skywaves. The input signal is fed to a tapped delay line with adjustable delays. The delayed signals at each tap are modulated in both amplitude and phase by a baseband tap gain function $Q_i(t)$. These delayed and modulated signals are next added to a noise function $V_n(t)$ to give the received signal. Each tap corresponds to a skywave that is resolvable in time. The noise function corresponds to any type of additive noise and, in certain cases interference, occurring on the channel, but is represented here as white Gaussian noise, for the reasons given in Section 2.4.5.

This model is an intuitively obvious model for an HF radio channel and remains a fairly general one until the tap gain functions $Q_i(t)$ are completely specified. In the particular case considered here, the tap gain functions are assumed to have the following properties [14,16].

1. Each tap gain function is a complex valued, zero mean, Gaussian random process.

2. Each tap gain function is independent of the other tap gain functions.

3. Each tap gain spectrum (generally) consists of the sum of two Gaussian functions of frequency.

The first of these hypotheses gives rise to Rayleigh fading being imposed on each skywave where the signal envelope follows a Rayleigh distribution and the phase is uniformly distributed. The second hypothesis ensures that the fading on each skywave is uncorrelated, and as a consequence of the zero mean criterion, is also independent. The third hypothesis reflects the fact that, included in each skywave are the two magnetoionic components, that is, the ordinary and the extraordinary rays. However, if the frequency spreads of these two components are assumed to be equal and significantly larger than their frequency shifts (as on the daytime F-layer mode [16]), then a single Gaussian function is sufficient to represent the tap gain spectrum.

The model used in this thesis assumes a single Gaussian function for each tap gain spectrum. Moreover, for further simplification, the average power and frequency spreads of the different skywaves themselves are assumed to be equal, and the frequency spreads are also assumed to be large, compared to the frequency shifts. Paradoxically, this simplified model, although obviously not representing the exact nature of the HF channel, does reflect the 'worst case' type situations that could be encountered. For example, the assumption that the average power in the different skywaves is equal, leads to the worst fades, since these occur when two skywaves are of equal strength and are in phase opposition. The 'equal frequency spread' assumption would not significantly affect the performance of the modem relative to its performance over a 'real' channel since the spreads being modeled are much larger than those observed on typical HF radio channels. The frequency shifts could also be assumed to be taken care of by these large frequency spreads, and thus need not be included as a seperate form of distortion. Moreover, even if there were an abnormally large shift in the mean carrier frequency, this would be detected and rectified by the carrier recovery circuit in the demodulator section of the modem.

2.6.2 Generation of the $Q_i(t)$

The tap gain functions $Q_i(t)$ must be generated within the constraints and specifications given in Section 2.6.1, in order that they impose Rayleigh fading on the input signal. Consider one such function $Q_1(t)$, or equivalently, one propagation path. The Rayleigh fading present on this path can be modeled as illustrated in Fig. 2.6.2 [17]. The input signal is passed through a 90° phase shifting network to give two signals in phase quadrature. This 90° phase shift should be carried out over the entire bandwidth of the input signal, or equivalently, a Hilbert transform [12] of the input signal should be performed. These two signals are next modulated by two random variables, $q_1(t)$ and $q_2(t)$,

where each of these variables are subjected to the following constraints.

1. Each must possess a zero mean, Gaussian probability density function with the same r.m.s. value.

2. Each must possess a Gaussian power spectrum with the same r.m.s. frequency.

3. The two processes must be uncorrelated.

The Gaussian power spectra of $q_1(t)$ and $q_2(t)$ are given by [17],

$$|q_1(f)|^2 = |q_2(f)|^2 = \exp\left(\frac{-f^2}{2f_{rms}^2}\right) \qquad (2.6.1)$$

The frequency spread introduced into an unmodulated carrier is the r.m.s. width of the power spectrum, and is given by

$$f_{sp} = 2f_{rms} \qquad (2.6.2)$$

The correlation between the fading rate and the r.m.s. frequency is given by [13],

$$\text{fading rate} = 1.475 f_{rms} \qquad (2.6.3)$$

where the fading rate is defined as the average number of downward crossings per unit time, of the signal through its median value.

The two processes $q_1(t)$ and $q_2(t)$ are themselves generated as shown in Fig. 2.6.3., by passing zero mean, white Gaussian noise with a variance of unity, through two seperate, but identical filters whose frequency responses are Gaussian. The filters chosen were 5-pole Bessel filters which meet the above requirements and their detailed design is given in Appendix A. The choice of Bessel filters was dictated by the fact that, as the order of these filters is increased, their frequency responses tend toward Gaussian. A block diagram of the filter used is given in Fig. A.1.

Nyquist theorem considerations dictate that the sampling frequency of the $q_i(t)$

in the digital implementation of the Bessel filter, should be more than twice the highest frequency contained in the $q_i(t)$. Since the filters have a Gaussian frequency response, the maximum frequency contained in the $q_i(t)$ is unbounded and could, theoretically, be considered to be infinity. However, for the highest frequency spread of interest, i.e., 2 Hz., it can be seen from Table A.1, that the cutoff frequency of the filter is 1.17 Hz., giving rise to a 3-dB bandwidth of 2.34 Hz. This implies that a sampling frequency of more than, say, 10 Hz., should be adequate for accurate representation of the filters in the digital domain.

However, the sampling rate of the fading processes $q_i(t)$ are ultimately determined by the transmission rate of the modem. For example, in a 2.4 kbaud modem, every signal sample should be modified by the fading, thus forcing the sampling frequency of the $q_i(t)$ to be 2400 Hz. or more. However, as the sampling frequency is increased, the filter poles in the z-plane tend to be pushed extremely close to the unit circle, and a high degree of accuracy is now required to define the tap gain coefficients. This problem is solved by having an intermediate sampling frequency for the $q_i(t)$ and obtaining the remaining samples by a process of linear interpolation. The sampling frequency used in this work was 100 Hz., that is, a sampling interval of 10 ms. (see Appendix A).

Finally, the constant G_D at the input to the Bessel filter (see Fig. A.1), is employed to change the variance of the $q_i(t)$ to their required values. The objective here is to ensure that the total mean power input to the model is equal to the total mean power output. Thus, the total variance of all the $q_i(t)$'s should be unity. Combined with the requirement that the variances of all the $q_i(t)$'s should be equal, this implies that in a 3-skywave channel, each $q_i(t)$ should have a variance of 1/6, whereas in a 2-skywave channel, the variance of each $q_i(t)$ should be 1/4. This is because six sequences ($q_1(t)$-$q_6(t)$) are required for a 3-skywave channel and four sequences ($q_1(t)$-$q_4(t)$) are required for a 2-skywave channel. The values of $(G_D)^{-1}$ which give the required variances in the $q_i(t)$ for Channels 1, 2, and 3 are, respectively, 19378, 15832 and 319330.

2.7 VERIFICATION OF CHANNEL MODEL

As mentioned earlier, it has been proven that the channel model given in the previous section is a fair, if not exact, representation of an HF radio channel [16,17].

However, it still needs to be confirmed, (to within a reasonable degree of accuracy) that the simulated model conforms to the model given in the previous section. Toward this end, a number of tests, (designed to check the accuracy), have been carried out on the simulated model, and these are now presented, along with their results.

Figs. 2.7.1 and 2.7.2 give the baseband power spectra of the Bessel filter (Fig. 2.6.3), for the two frequency spreads of interest. These spectra should, ideally, be Gaussian in shape and as such, they are compared with the theoretical Gaussian spectrum obtained via eqn. 2.6.1. It could be seen from Figs. 2.7.1 and 2.7.2 that, in both cases, the simulated Bessel filter response is exactly coincident with the Gaussian response, in the range of their respective frequency spreads, namely 2 Hz. and 1 Hz. (This range is marked by the dashed lines in the above mentioned figures). As such, it could safely be assumed that the power spectra of the $q_i(t)$ are Gaussian.

Figs. 2.7.3 and 2.7.4 show, for the three channels, the cumulative distribution function of the envelope of the first skywave. For a Rayleigh fading channel, these functions should, of course, follow a Rayleigh cumulative distribution function and hence, are compared with such. The theoretical Rayleigh cdf (cumulative distribution function) is easily obtained by integrating the Rayleigh probability density function, that is, if the Rayleigh cdf is given by F(x), then,

$$F(x) = \int_0^x \frac{x}{\sigma^2} \exp\left(\frac{-x^2}{2\sigma^2}\right) dx$$

$$= 1 - \exp\left(\frac{-x^2}{2\sigma^2}\right) \qquad (2.7.1)$$

where σ^2 is the variance of the individual $q_i(t)$ (see Figs. 2.6.2 and 2.6.3).

Since the Rayleigh cdf is dependent on the variance of the individual $q_i(t)$, and since this variance itself is different for the 2-skywave and 3-skywave channels, (these variances are 1/4 and 1/6, respectively), the comparison for the 2-skywave channels should be seperate from the comparison for the 3-skywave channel. Hence the need for Figs. 2.7.3 and 2.7.4, where the former compares the cdf of the envelope of the first skywave in Channel 1 with the Rayleigh cdf corresponding to a 3-skywave channel, and the latter compares the cdf of the envelope of the first skywave in Channels 2 and 3 with the

Rayleigh cdf corresponding to a 2-skywave channel.

It can be seen, from these two figures, that the first skywave in all three channels exhibit Rayleigh statistics. The slight deviation which exists between the theoretical and simulated curves in Fig. 2.7.3 is, of course, due to the statistical nature of the simulation, where the value of σ^2 in the simulation, may not always be equal to 1/6 (or, for a 2-skywave channel, to 1/4). Even after these deviations, the general shape of the cdf is similar to that of a Rayleigh cdf, and as such, it could be assumed that all three channels exhibit Rayleigh fading.

Another important parameter in a Rayleigh fading channel is the distribution of the phase of the faded signal. As stated before, this phase should be uniformly distributed. Fig. 2.7.5 compares the cdf of the phase of the first skywave in the three channels with the cdf corresponding to a uniform distribution. In the case of every channel, there is an excellent correspondence between the theoretical and simulated curves, and as such, the phase of the first skywave in all three channels could be assumed to be uniformly distributed between $-\pi$ and $+\pi$.

Since the second and third skywaves are obtained via the same filter in the same manner as the first skywave, (apart from a change in value of a seed integer which initializes the random number generators), they exhibit the same statistics as the first skywave. Thus, all skywaves in the simulated HF radio channel model exhibit a Rayleigh distribution in amplitude and a uniform distribution in phase.

Figs. (2.7.6 - 2.7.8) show, for the 3 channels, the baseband, complex envelope of the first skywave when the input is considered to be an unmodulated sine-wave carrier. These skywaves are shown relative to their median values, where the median values are directly obtained from their respective cumulative distribution functions given in Figs. 2.7.3 and 2.7.4 (The median value is the point at which the cdf is 0.5, or 50%). It should be emphasised that Figs. (2.7.6 - 2.7.8) show the fading observed on the first skywave and not the resultant fading due to all the skywaves at the output of the channel model. As such, no remarks can be made, at this stage, about the overall fading structure of the channel. Nevertheless, these figures do give some indication of the number of fades likely to be encountered in the respective channels. In a 25 sec. time period, the first skywave in Channels 1, 2 and 3 exhibits, respectively, 37, 36 and 14 fades, where a fade is regarded as a downward crossing of the signal through its median value. There are a fewer number of fades in the first skywave in Channel 3, when compared with the first skywave in Channels 1 and 2. This is to be expected, since the fading rate depends on the

frequency spread (eqns. 2.6.2 and 2.6.3), and the frequency spread in Channel 3 (1 Hz.) is half that in Channels 1 and 2 (2 Hz.). In terms of the depths of the fades, there is at least one fade in excess of - 30 dB. These depths however, are expected to change when the other skywaves are added to form the resultant HF channel model. This point is explained further in Section 4.6.

Finally, Figs. (2.7.9 - 2.7.11) show the phase variations of the first skywave in Channels 1, 2 and 3. It could be seen that in Channel 3, (the channel with the lower frequency spread of 1 Hz.), the phase variations are slower than in the other two channels. In all three channels however, the phase varies between $-\pi$ and $+\pi$. This is, as proven earlier, to be expected.

References - Chapter 2

1. **Picquenard, A.,** 'Radio wave propagation', Macmillan Press, 1974.

2. **David, P. and Voge, J.,** 'Propagation of waves', Permagon Press, 1969.

3. **Davies, K.,** 'Ionospheric radio propagation', National Bureau of Standards, 1965.

4. **Budden, K.G.,** 'Radio waves in the ionosphere', Cambridge University Press, 1961.

5. **Bain, W.C. and Rishbeth, H.,** 'Developments in ionospheric physics since 1957', The Radio and Electronic Engineer, vol. 45, no. 1 / 2, pp. 3 - 10, January / February 1975.

6. **Goldberg, B.,** '300 KHz. - 30 MHz. MF / HF', IEEE Trans. on Communication Technology, vol. COM-14, no. 6, pp. 767 - 784, December 1966.

7. **CCIR,** 'Ionospheric properties', XVth Plenary Assembly CCIR, vol. VI, Report 725 - 1, pp. 1-15, 1982.

8. **Rush, C.M.** 'Ionospheric radio propagation models and predictions - A mini review, IEEE Trans. on Antennas and Propagation, vol. AP-34, no. 9, pp. 1163 - 1170, September 1986.

9. **Maslin, N.M.,** 'High data rate transmission over HF radio links', The Radio and Electronic Engineer, vol. 52, no. 2, pp. 75 - 87, February 1982.

10. **CCIR,** 'Multipath propagation on HF radio circuits', Xth Plenary Assembly CCIR, vol VI, Report 203, pp. 49 - 51, 1963.

11. **Stein, S.,** 'Fading channel issues in system engineering', IEEE Journal on selected areas in Communications, vol. SAC -5, no. 2, pp. 68 - 89, February 1987.

12. **Proakis, J.G.,** 'Digital communications', McGraw Hill, 1983.

13. **CCIR,** 'Ionospheric propagation characteristics pertinent to radiocommunication systems design - (fading)', XIIIth Plenary Assembly CCIR, vol.VI, Report 266-3, pp. 207 - 217, 1974.

14. **CCIR,** 'HF ionospheric channel simulators', XIIIth Plenary Assembly CCIR, vol. III, Report 549-1, pp. 47 - 52, 1974.

15. **Clark, A.P.,** 'Principles of digital data transmission', 2nd Edition, Pentech Press, 1983.

16. **Watterson, G.C., Juroshek, J.R. and Beusema, W.D.,** 'Experimental confirmation of an HF channel model', IEEE Trans. on Communication Technology, vol. COM-18, no. 6, pp. 792 - 803, December 1970.

17. **Ralphs, J.D. and Sladen, F.M.E.,** 'An HF channel simulator using a new Rayleigh fading method', The Radio and Electronic Engineer, vol. 46, no. 12, pp. 579 - 587, December 1976.

	IONOSPHERIC LAYER			
	D	E	F	
			F_1	F_2
Virtual height (km)	50 - 90	90 - 130	130 - 210	250 - 350
Electron density (m^{-3})	10^9	10^{11}	$2 * 10^{11}$	10^{12}
Critical frequency (MHz.)	0.5	4	5	10

Table 2.2.1 Main parameters of ionospheric layers

	Multipath Spread (ms.)	Frequency Spread (Hz.)
GOOD	0.5	0.1
MODERATE	1	0.5
POOR	2	1
FLUTTER	0.5	10

Table 2.5.1 **CCIR classification of HF channels**

	Multipath Spread (ms.)	Frequency Spread (Hz.)
CHANNEL 1	1.1 & 3	2
CHANNEL 2	3	2
CHANNEL 3	2	1

Table 2.5.2 **Channels used in tests**

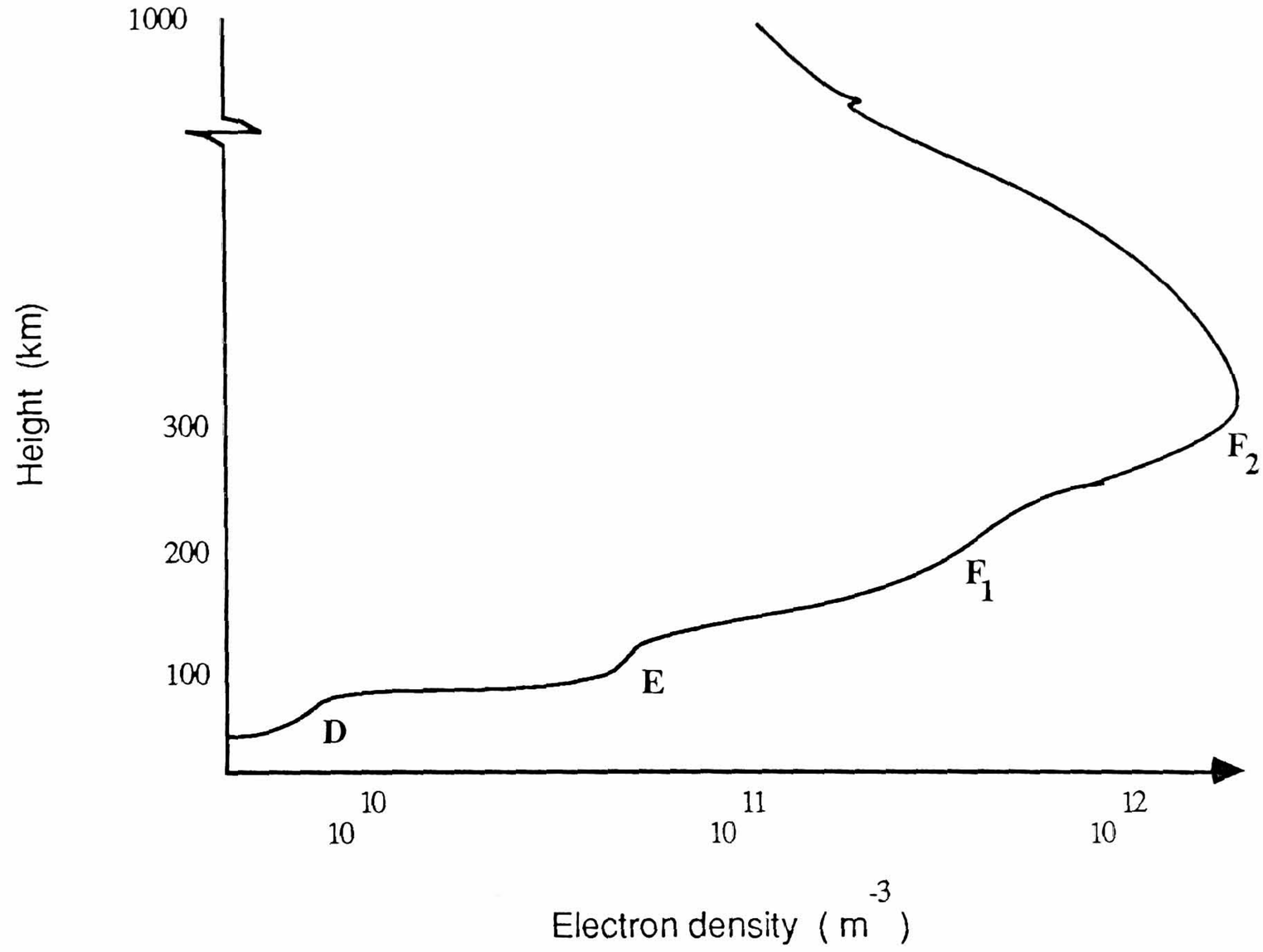


Fig. 2.2.1 **Electron density profile**

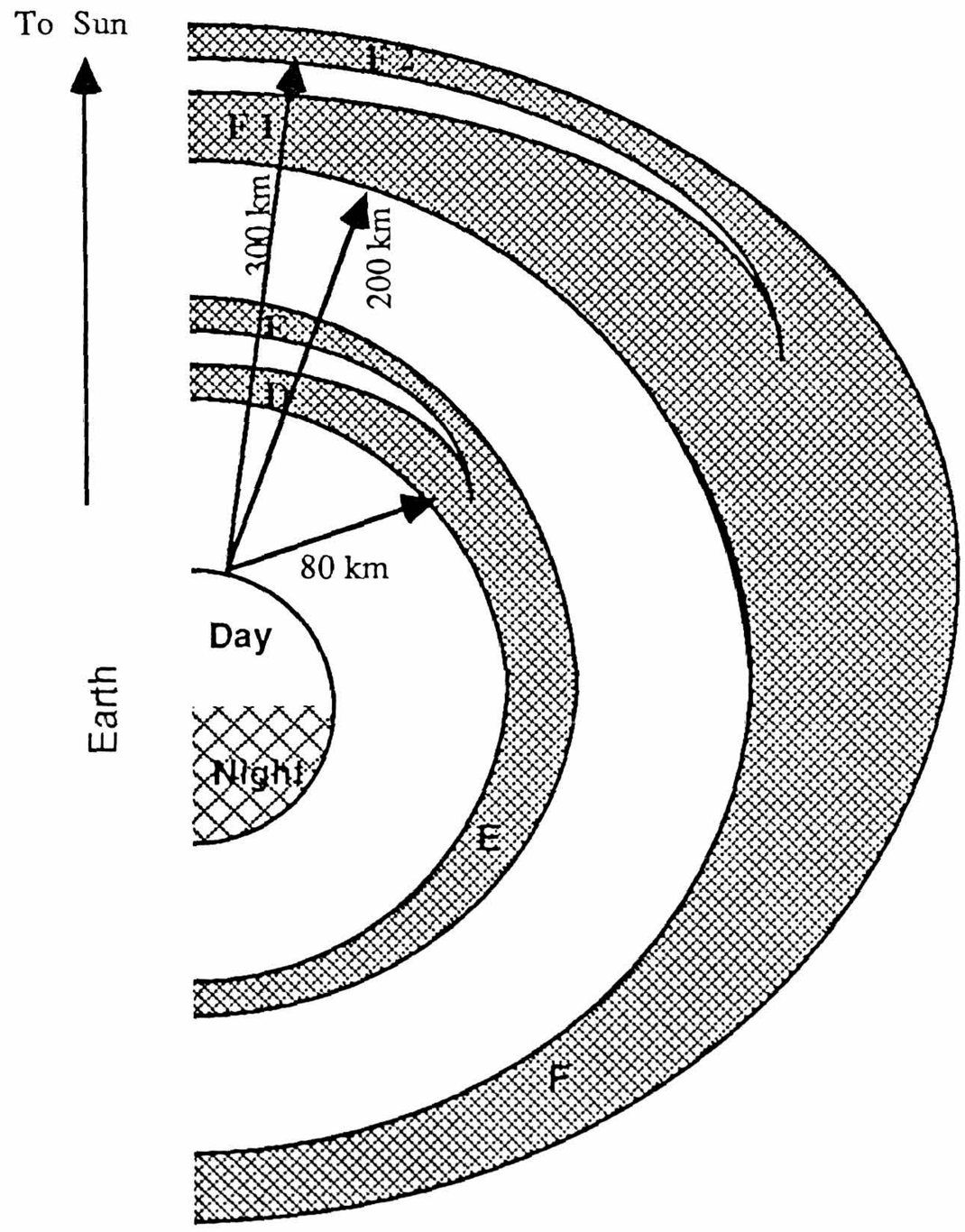


Fig. 2.2.2 **Ionospheric regions as a function of height**

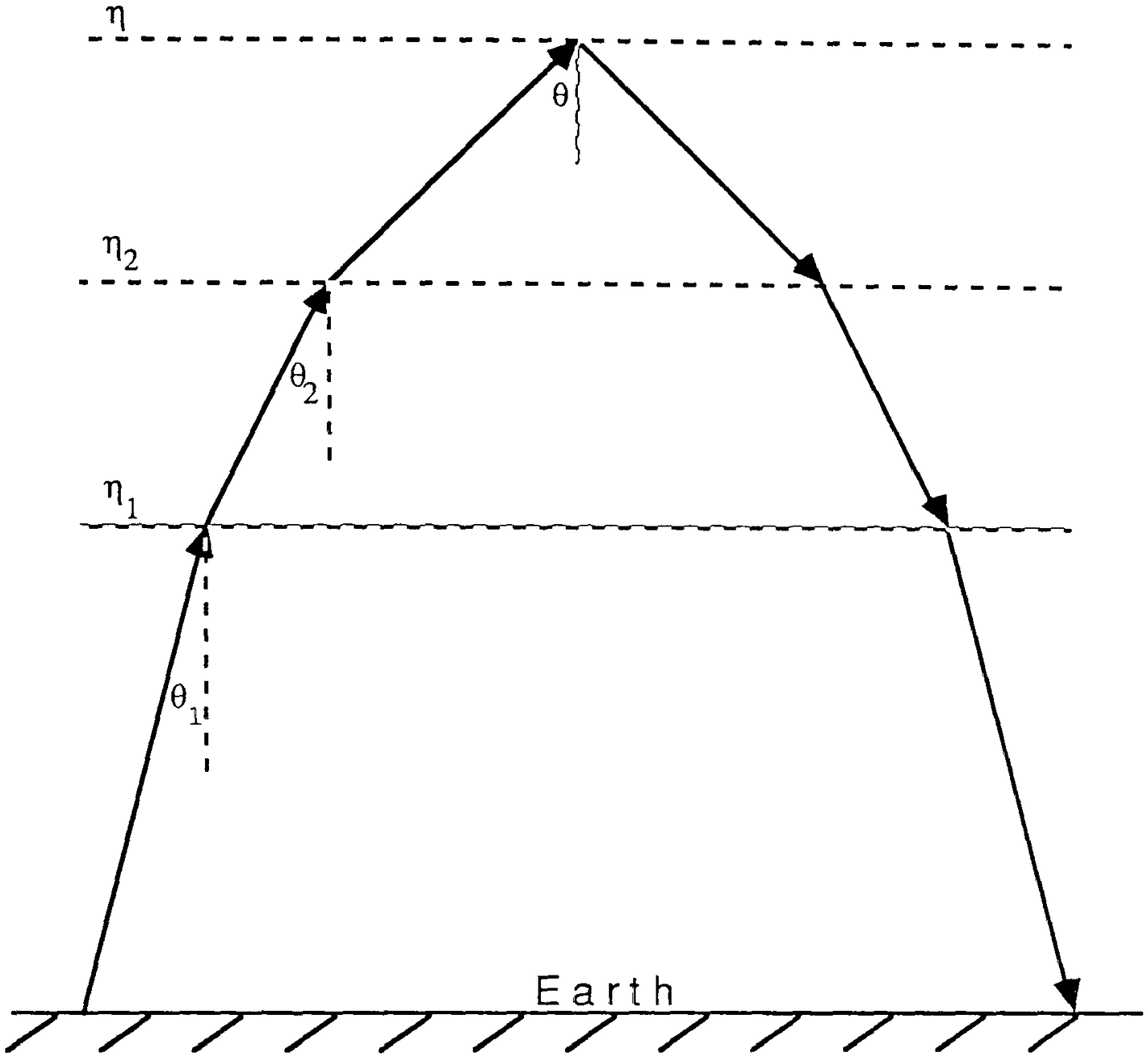
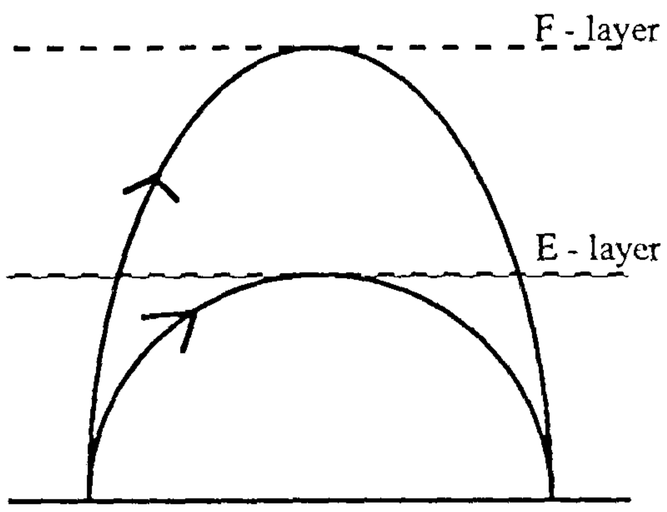
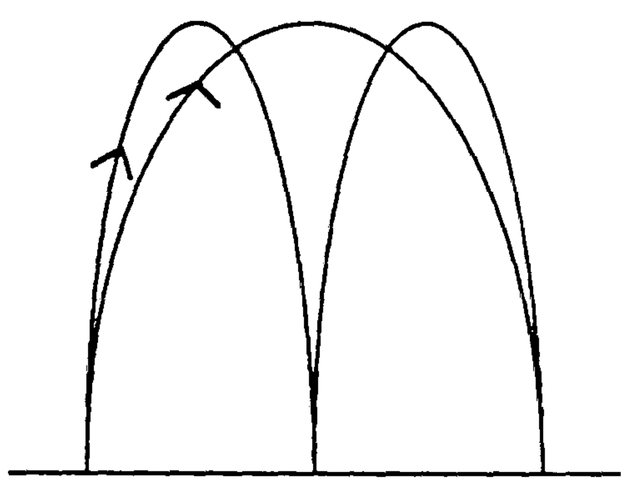


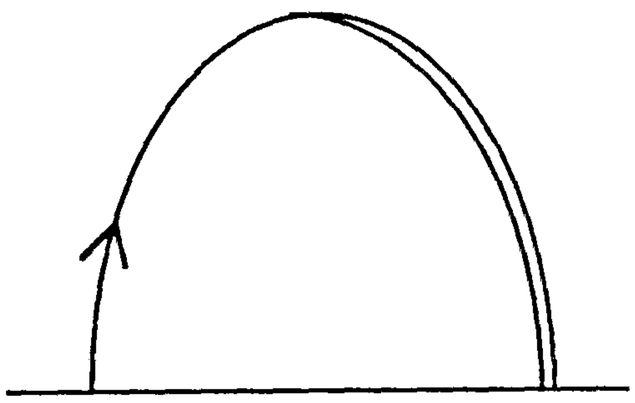
Fig. 2.3.1 Progressive refraction of a radio wave



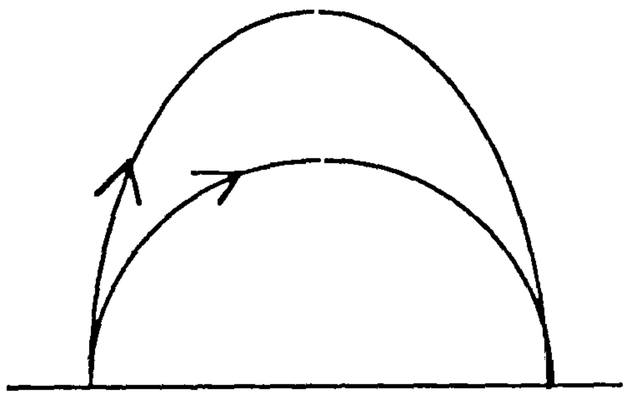
different layers



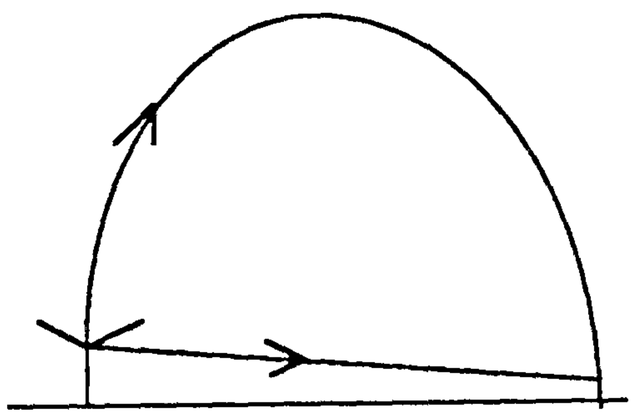
different hops



magnetoionic splitting



high / low angle



skywave / groundwave

Fig. 2.4.1 **Different modes of skywave propagation**

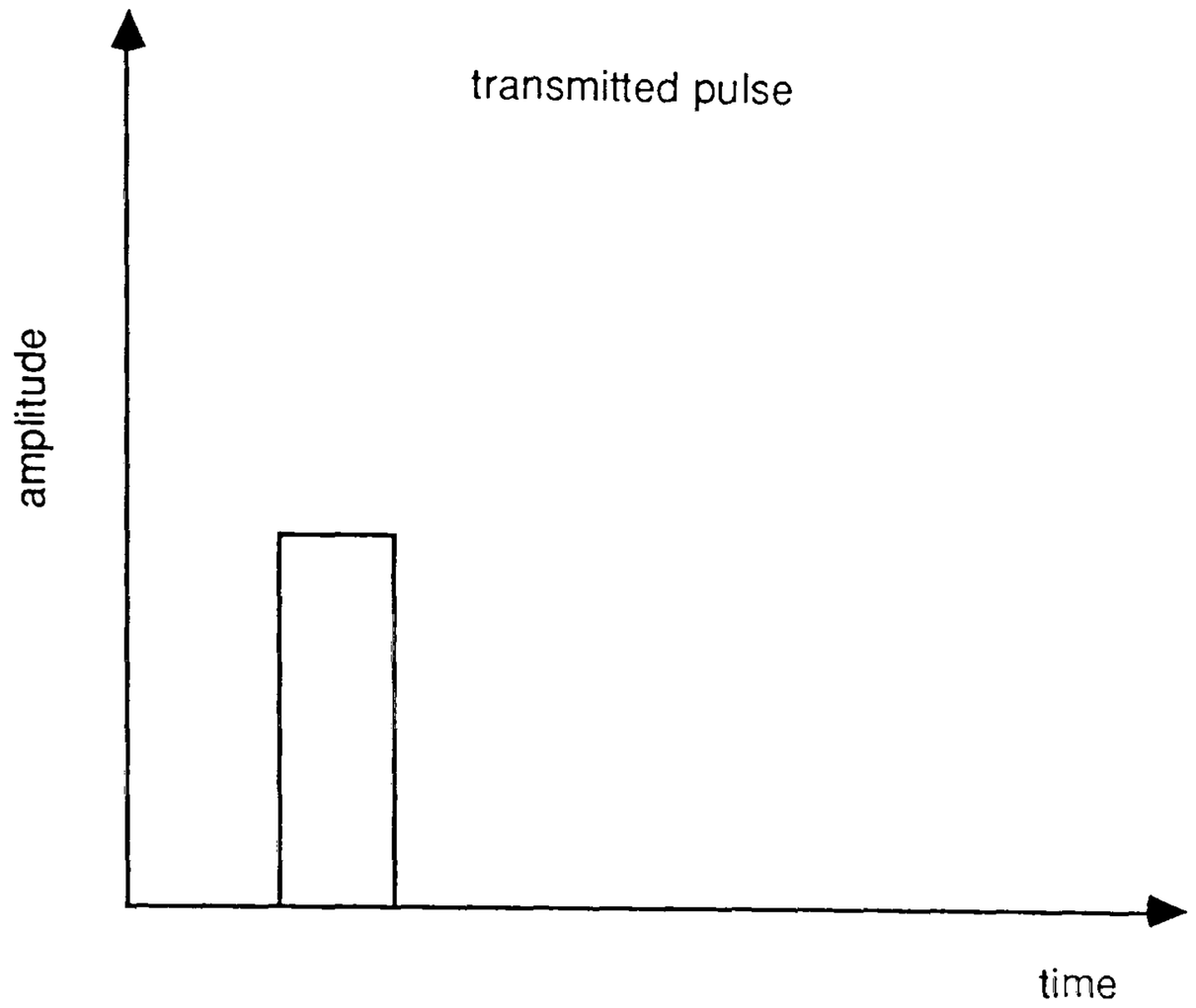
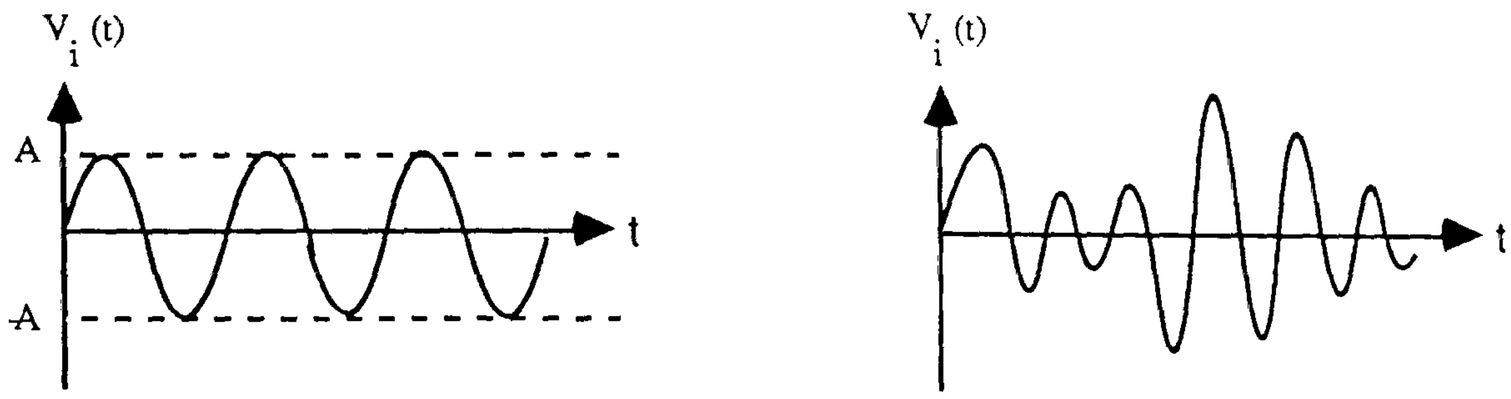
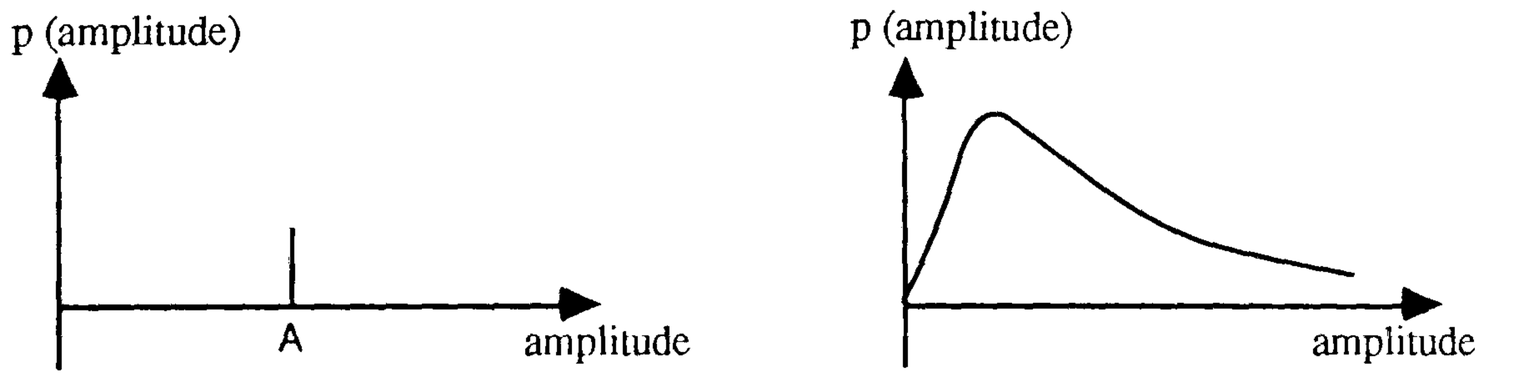


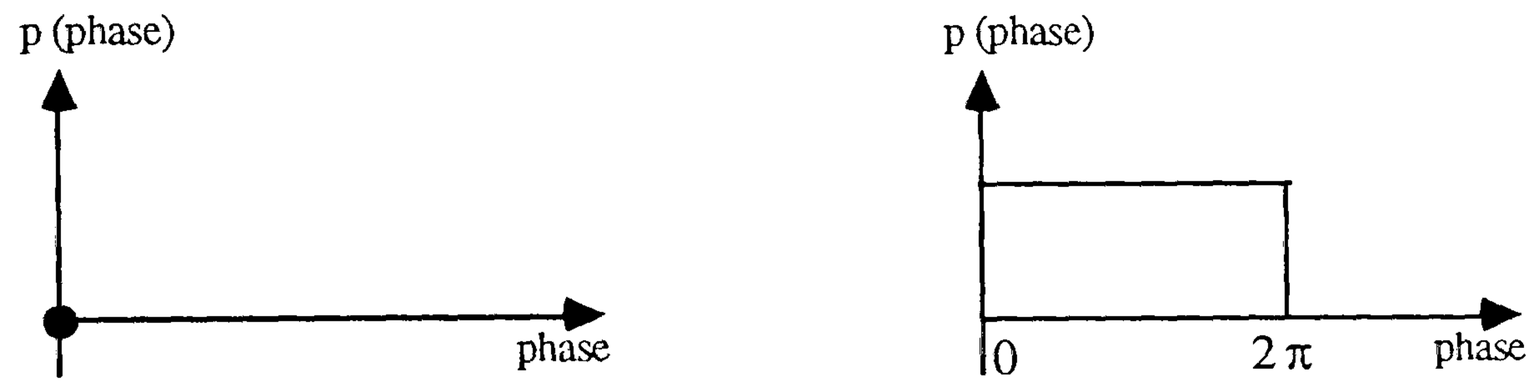
Fig. 2.4.2 Response of a multipath channel



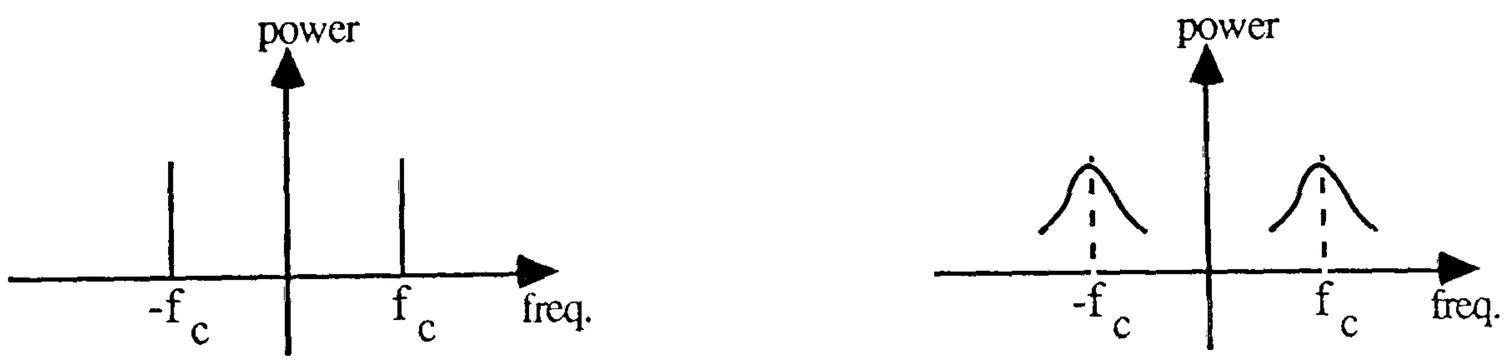
a) signal waveform



b) p.d.f. of amplitude



c) p.d.f. of phase



d) power spectrum

Characteristics of unmodulated carrier

Characteristics of Rayleigh faded signal

Fig. 2.4.3 Effects of signal transmission over a Rayleigh fading channel

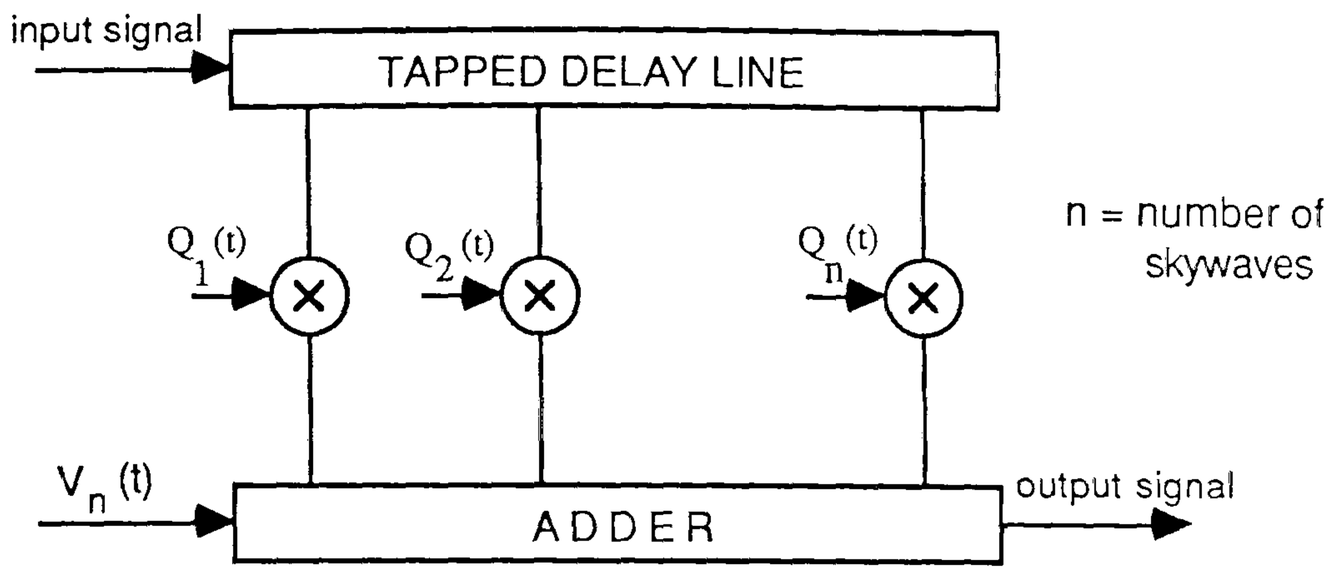


Fig. 2.6.1 **Model of HF channel**

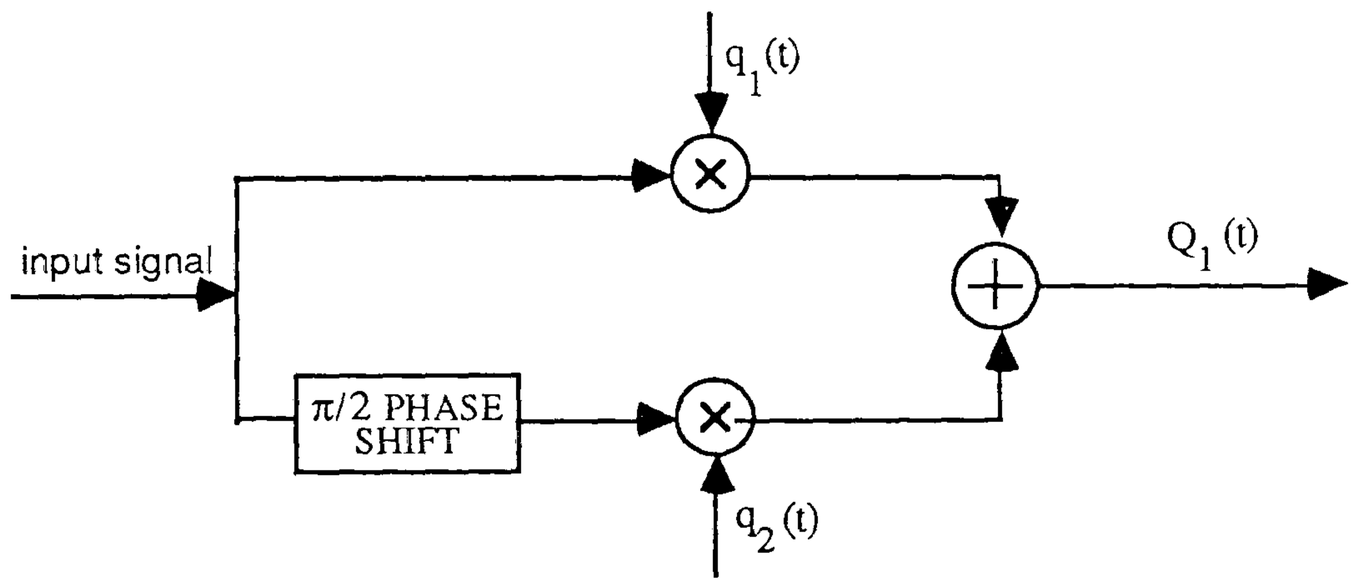


Fig. 2.6.2 **Generation of a single Rayleigh fading skywave**

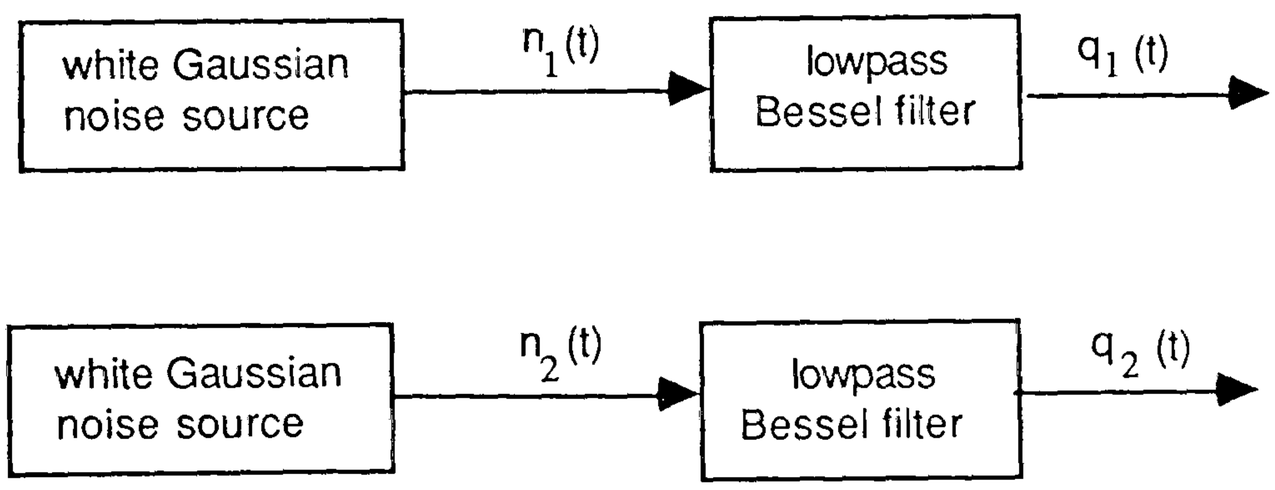


Fig. 2.6.3. **Generation of $q_1(t)$ and $q_2(t)$**

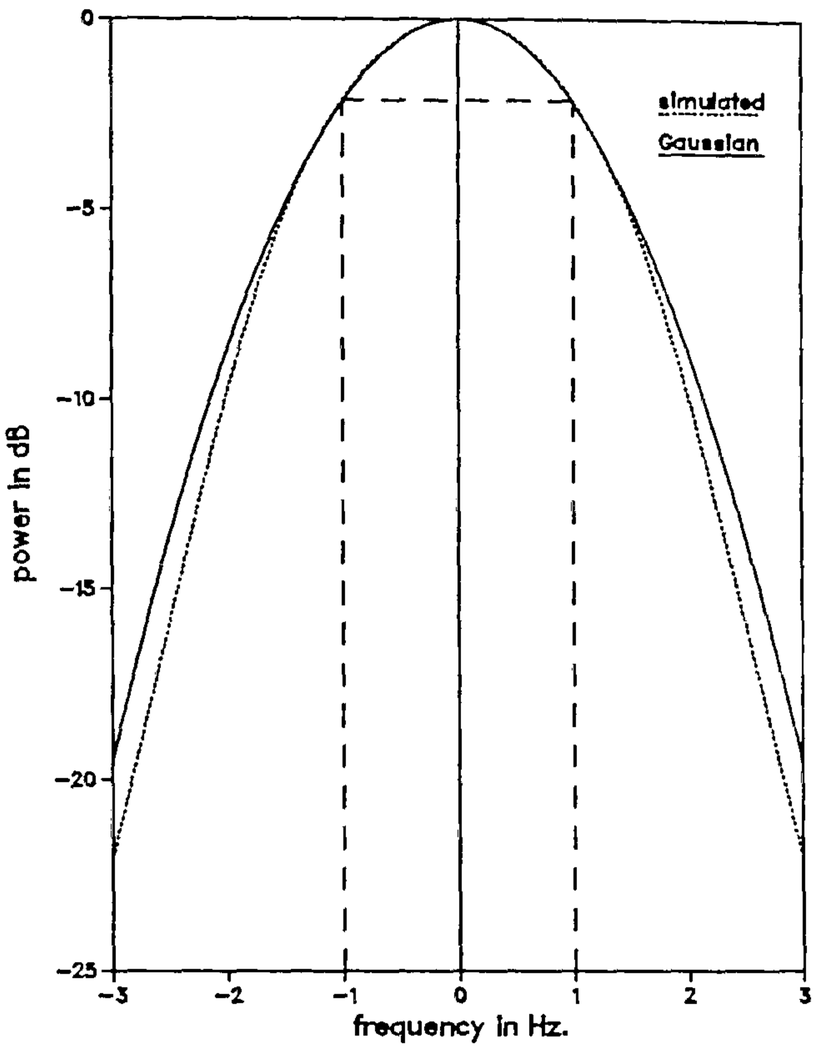


Fig.2.7.1. Power spectrum of Bessel filter for fsp=2Hz.

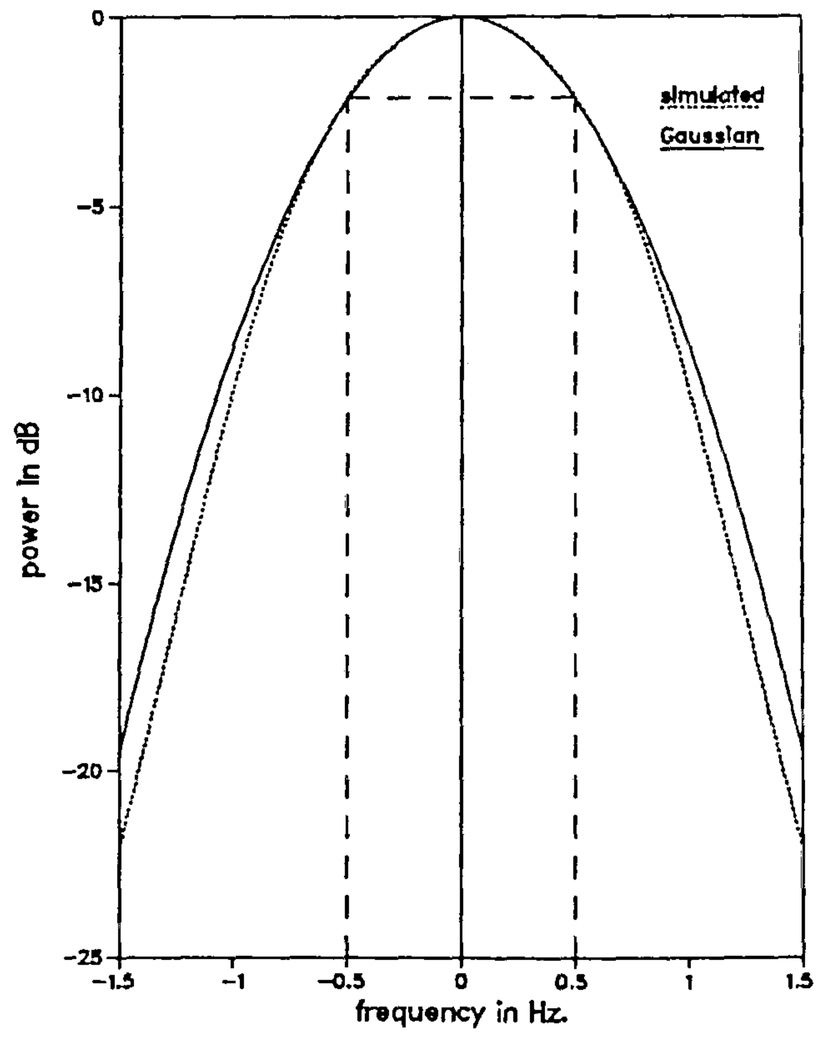


Fig.2.7.2. Power spectrum of Bessel filter for fsp=1Hz.

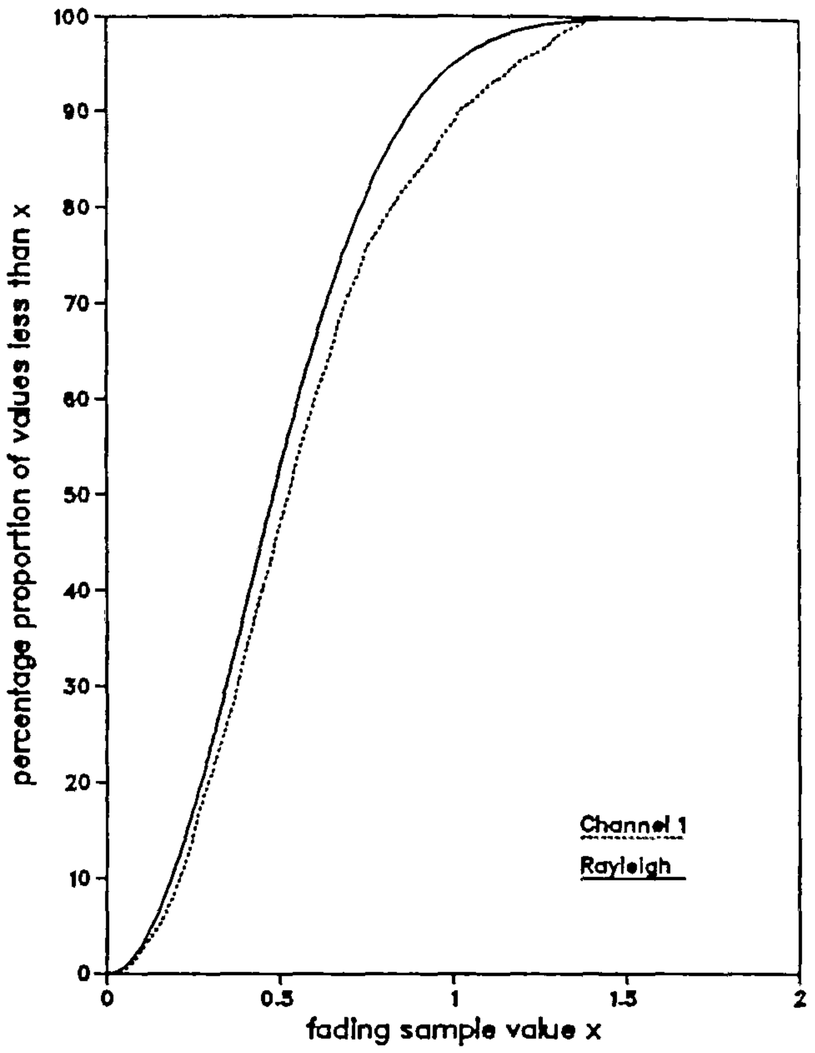


Fig.2.7.3. CDF of fading amplitude for single skywave in Channel 1

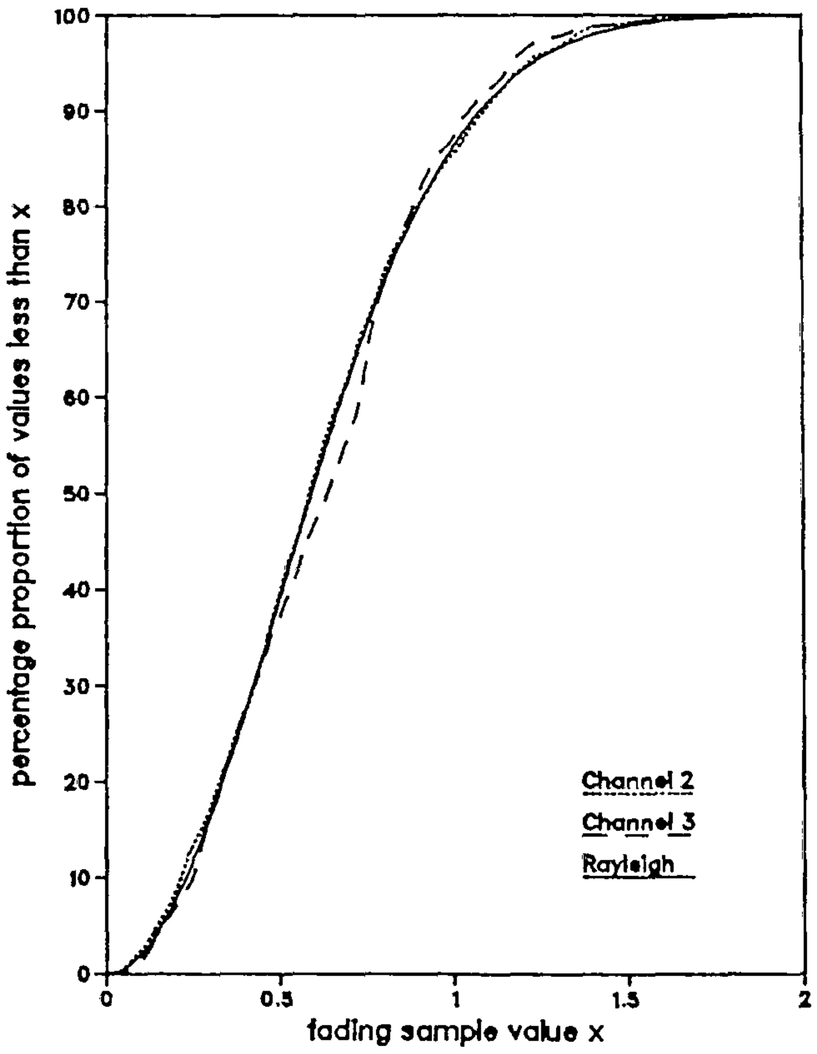


Fig.2.7.4. CDF of fading amplitude for single skywave in Channels 2 and 3

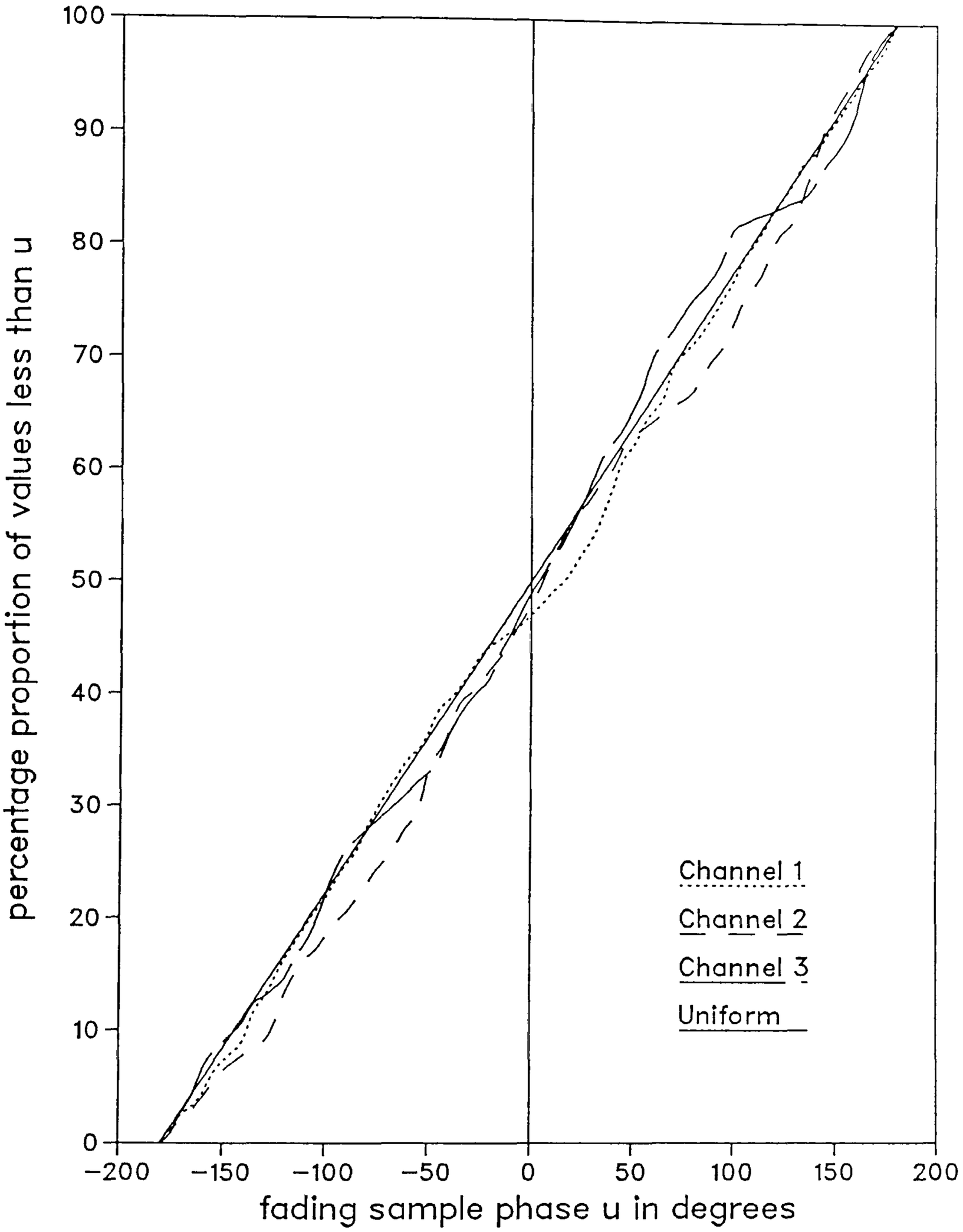


Fig.2.7.5. CDF of fading phase for single skywave in Channels 1 , 2 and 3

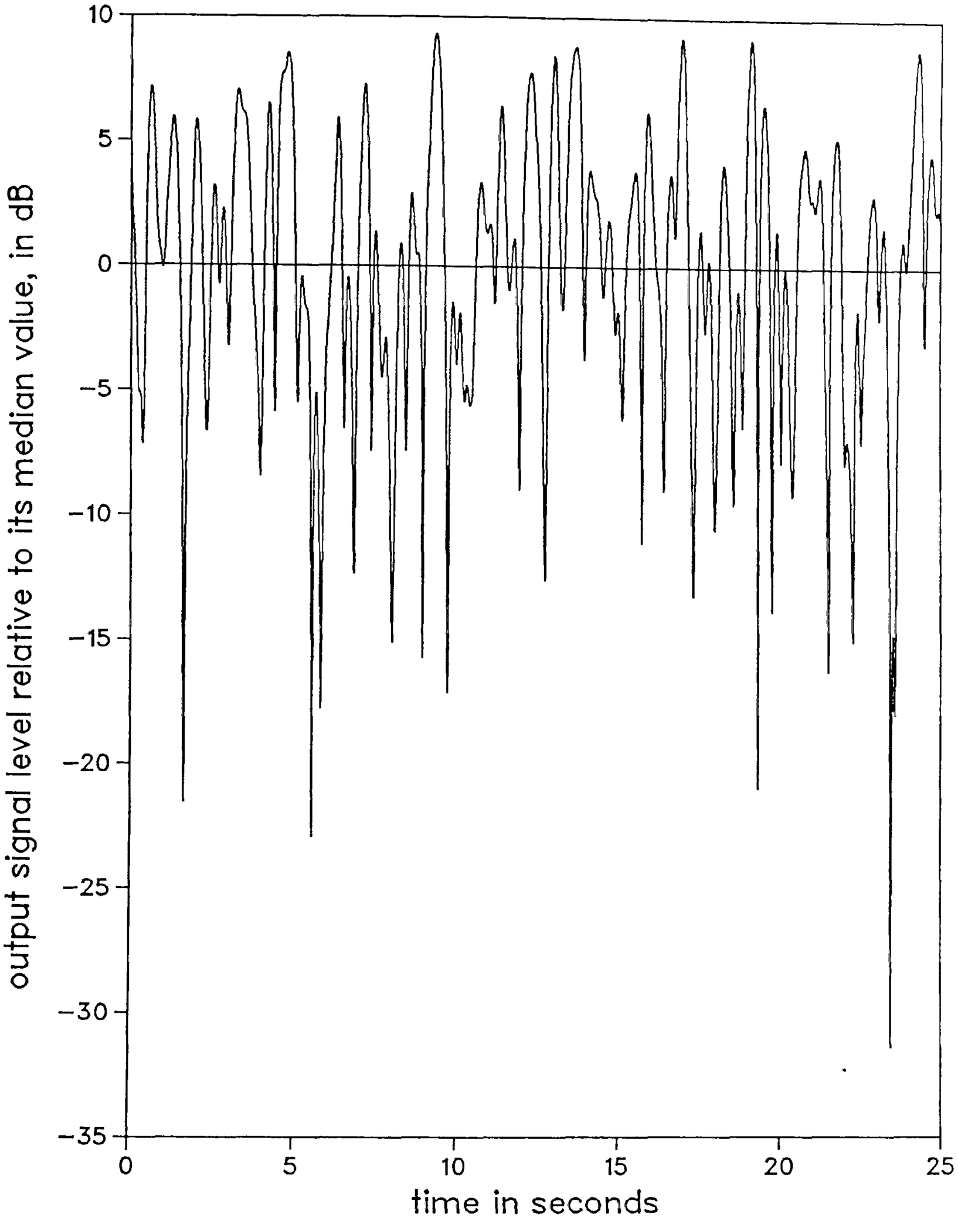


Fig.2.7.6. Rayleigh faded skywave in Channel 1

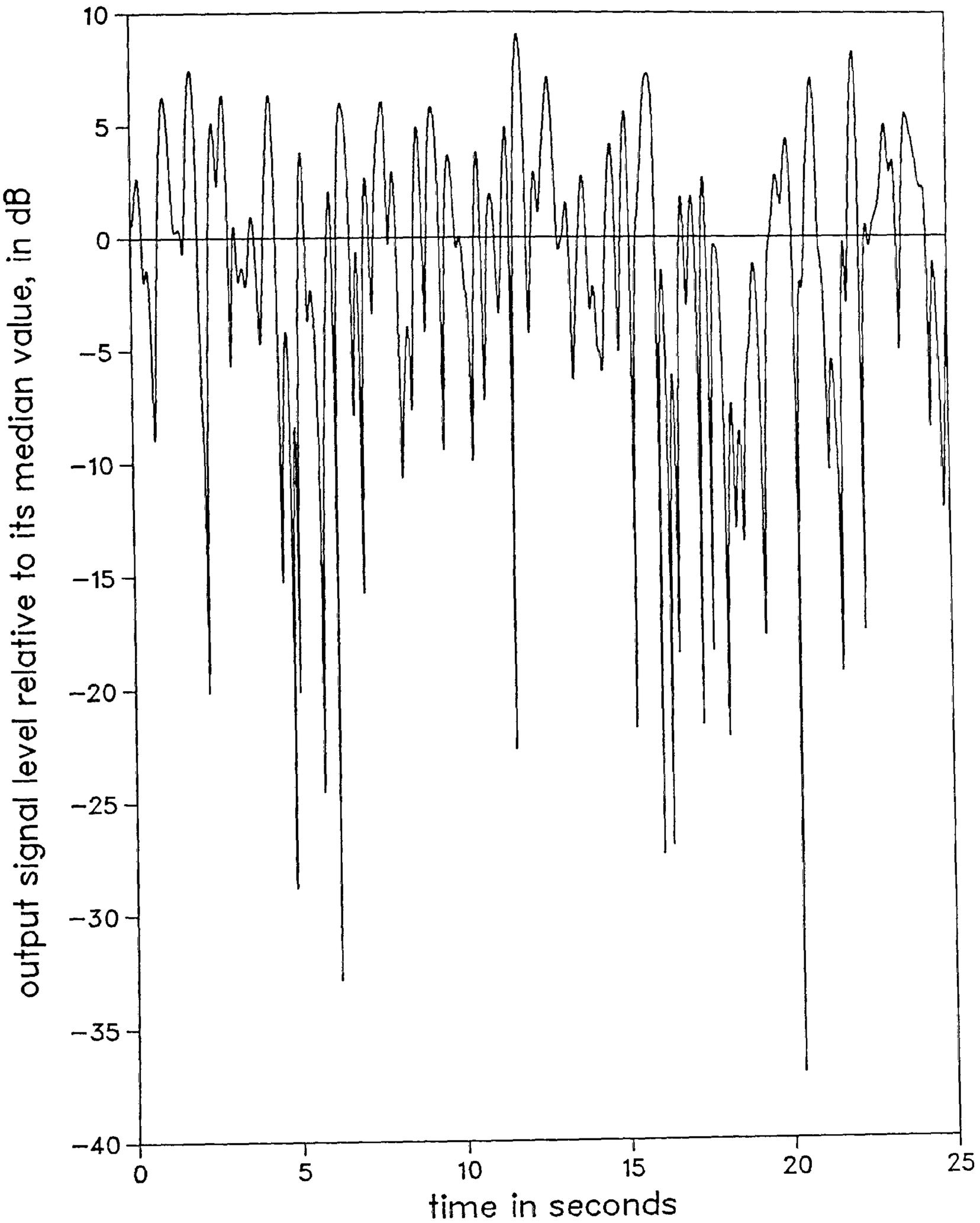


Fig.2.7.7. Rayleigh faded skywave in Channel 2

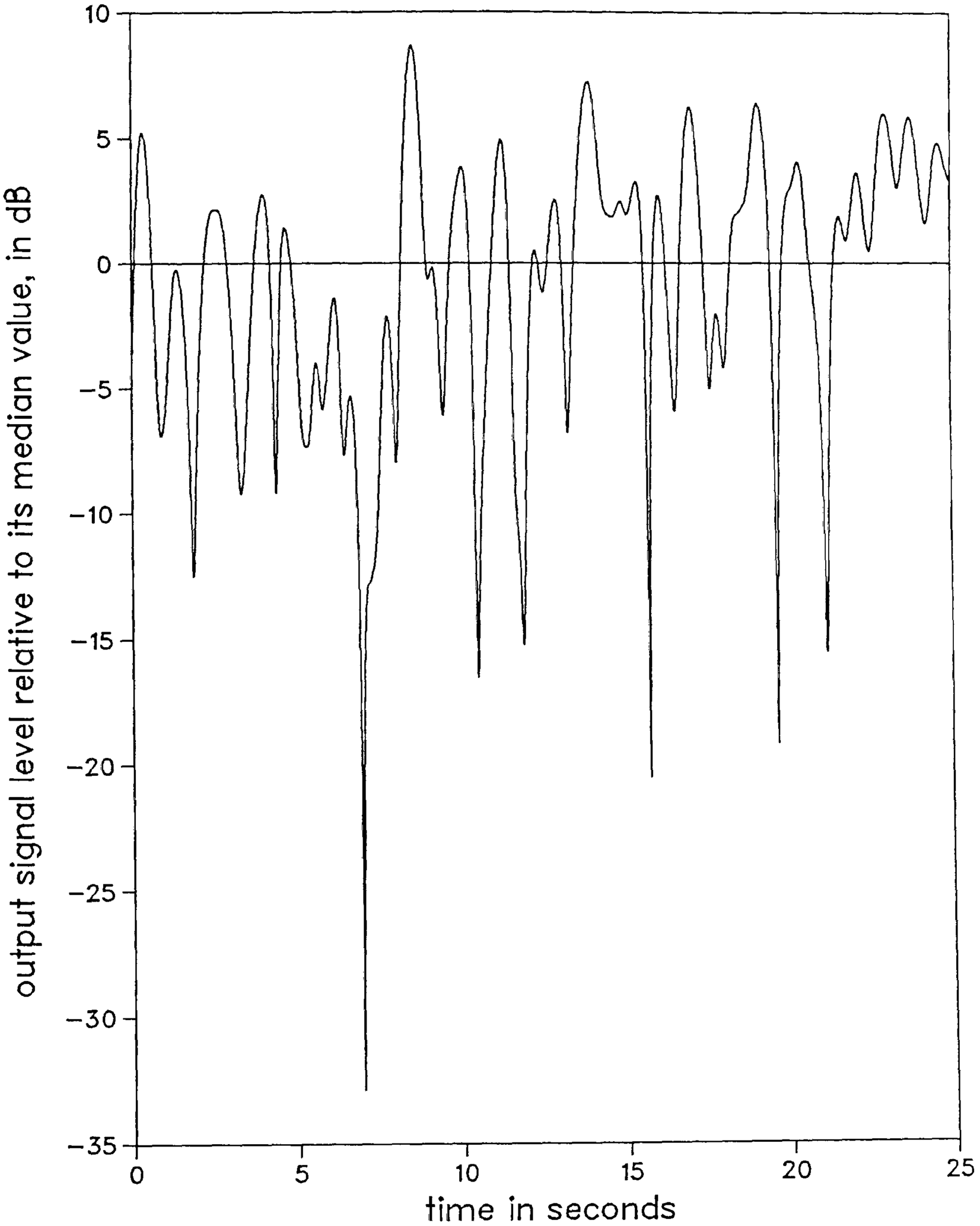


Fig.2.7.8. Rayleigh faded skywave in Channel 3

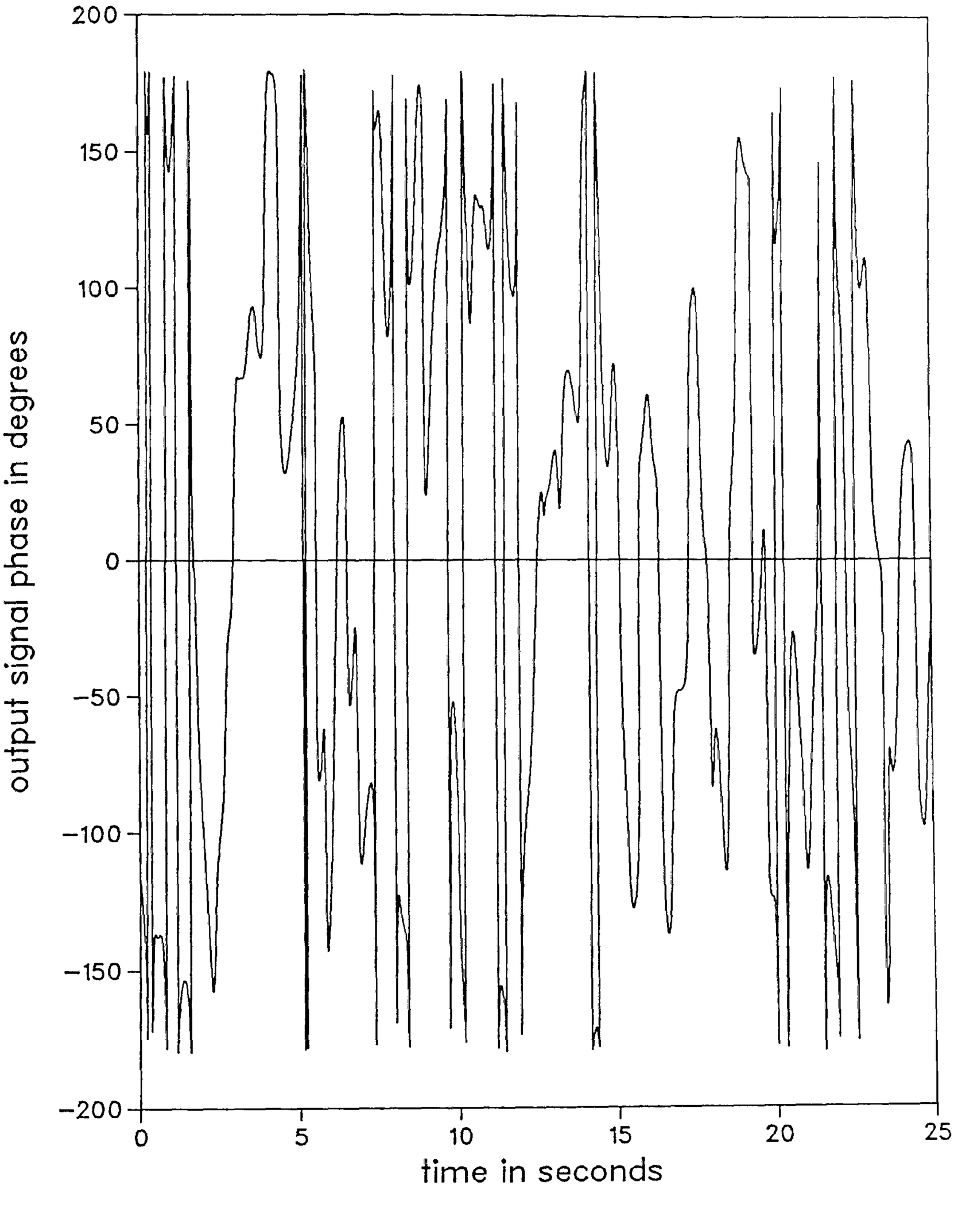


Fig.2.7.9. Phase variation of skywave in Channel 1

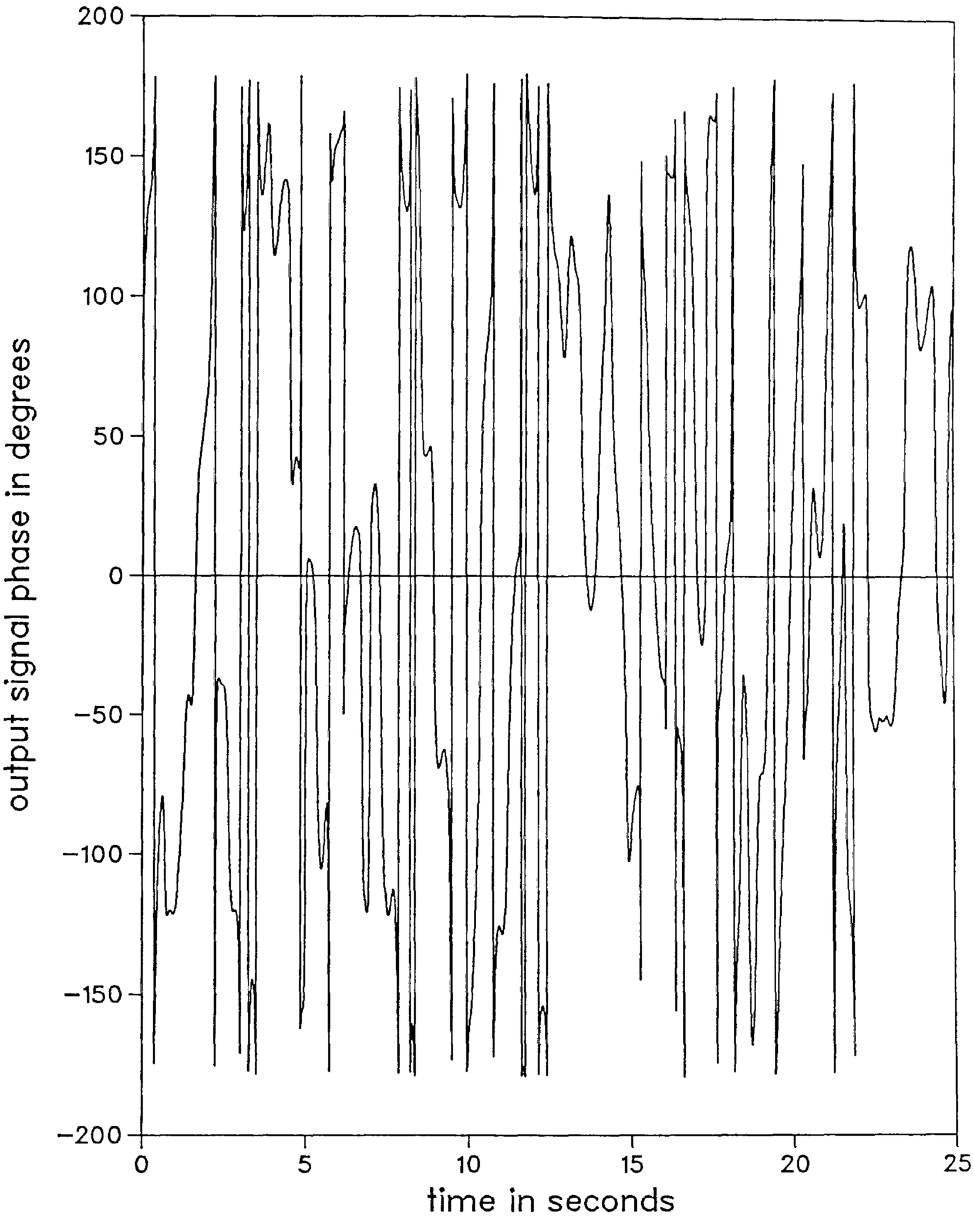


Fig.2.7.10. Phase variation of skywave in Channel 2

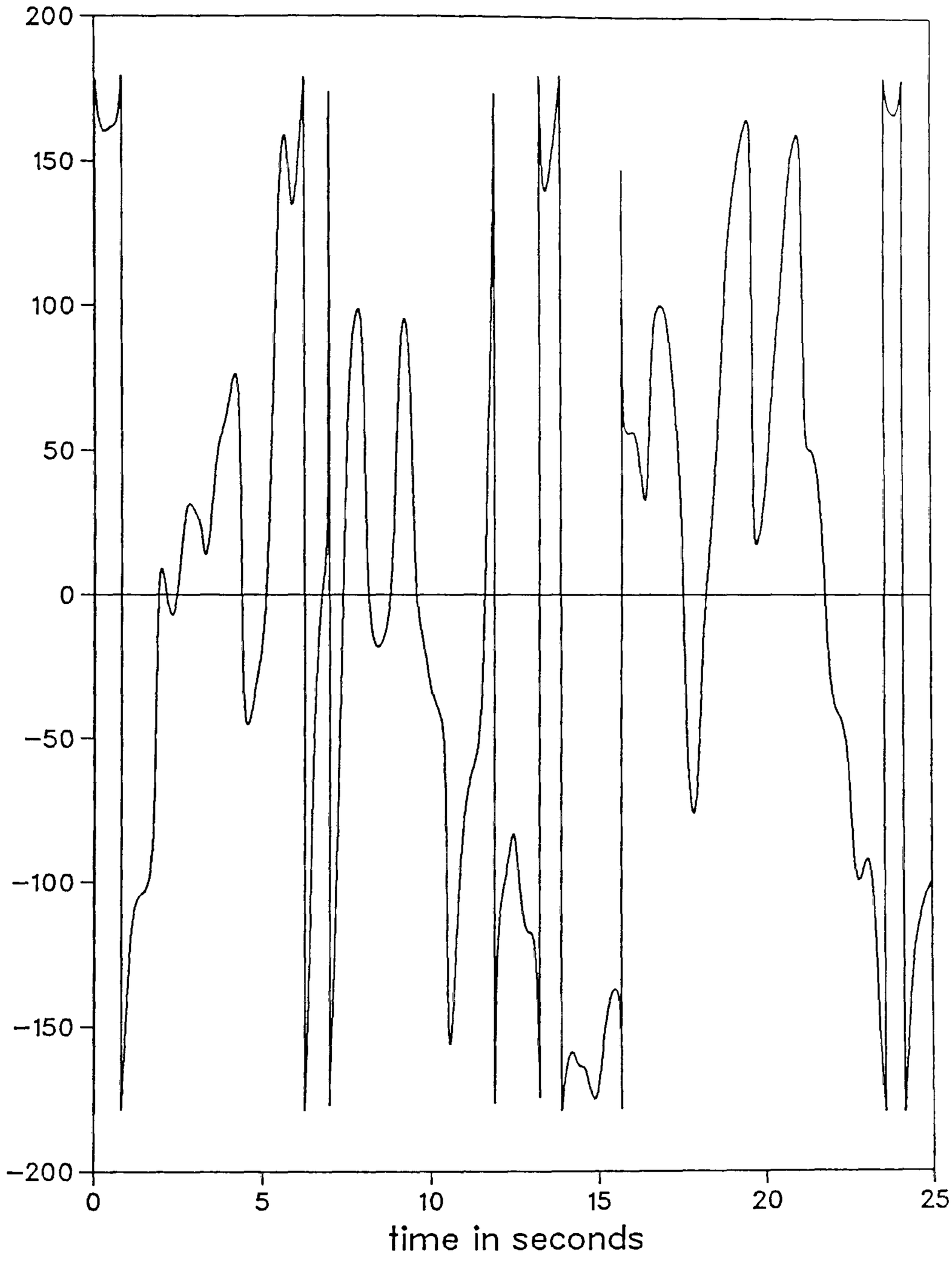


Fig.2.7.11. Phase variation of skywave in Channel 3

CHAPTER 3

DATA TRANSMISSION USING QUADRATURE AMPLITUDE MODULATION

3.1 INTRODUCTION

Quadrature amplitude modulation (QAM) is a multilevel, digital modulation scheme used to modulate information bearing data signals, so that they become compatible for transmission over practical bandpass channels. It employs two double sideband surpressed carrier AM signals of the same carrier frequency for this purpose, where these are in phase quadrature and are transmitted simultaneously. The spectrum of a QAM signal is, of course, similar to that of a double sideband surpressed carrier AM signal, with the upper and lower sidebands centred around the carrier frequency.(1)

A distinct advantage of QAM is that it is a highly bandwidth efficient modulation scheme which is also linear [1-3]. A QAM signal has m levels where $m \geq 4$. By increasing the number of levels, it is possible to increase the information transmission rate over a given bandwidth. This scheme of things cannot, of course, be carried on ad infinitum since the increase in the number of levels is also accompanied by more acute intersymbol interference problems and timing and phase jitter problems [1]. However, a profitable compromise is achieved in most situations. Moreover, the modulation (and demodulation) process is linear, thus greatly simplifying the theoretical analysis of the system by reducing it to that of a linear baseband channel [3-5]. These, and other factors [6] combine to make QAM one of the most widely used modulation methods in both radio and satellite systems.

Since a thorough and detailed discussion into every aspect of QAM is beyond the scope of the thesis, this chapter will be confined to developing a theoretical model for the QAM system which lends itself into the more detailed model for data transmission using QAM over HF radio links.

3.2 THE QAM SYSTEM MODEL

A QAM data transmission system is shown in Fig. 3.2.1. The $\{s_{1,i}\}$ and $\{s_{2,i}\}$ are two streams of data symbols which carry the information to be transmitted. They are

statistically independent and each symbol is equally likely to have any one of $\log_2 m$ possible values. An example of the possible values of $s_{1,i}$ and $s_{2,i}$ is given in Fig. 3.2.2. This is normally referred to as the QAM signal constellation. These symbols are assumed to be transmitted in the form of impulses at every iT seconds.

These data symbols are initially passed, seperately, through two lowpass transmitter filters (A_1 and A_2), having the same impulse response a(t), and transfer function A(f). a(t) here is real valued. A_1 and A_2 are signal spectrum shaping filters which limit the bandwidth of the resultant QAM signal to that available on the transmission path. In the case considered here, |A(f)|, as shown in Fig. 3.2.3, is assumed to be bandlimited from -1/2T to 1/2T Hz., thus ensuring that the signaling rate of 1/T Hz. is at the Nyquist rate [1].

The two signals at the outputs of A_1 and A_2 are next modulated by two carriers which are in phase quadrature. The carrier frequency, f_c, is chosen such that the spectra of the baseband signals are shifted into the passband of the transmission path. These modulated carrier signals are now added together, to form the resultant QAM signal x(t), whose amplitude spectrum, |X(f)|, is as shown in Fig. 3.2.4.

The QAM signal, x(t), is transmitted over a channel whose impulse response is a real valued function h(t). The channel is assumed to introduce stationary, white Gaussian noise, which is added at the output of the transmission path. The noise function, n(t), is a Gaussian random process with zero mean and a two-sided power spectral density of $N_0/2$.

At the receiver, the noisy QAM signal is first passed through a bandpass receiver filter C, whose function is to remove any noise components lying outside the signal frequency band. Toward this end, the filter has a bandlimited amplitude spectrum |C(f)|, where its value is unity for $f_c - 1/2T \le |f| \le f_c + 1/2T$, and zero elsewhere, and is as shown in Fig. 3.2.5. The impulse response of this filter is c(t) and is real valued. The signal at the output of the receiver filter, z(t), is coherently demodulated by two reference carriers in phase quadrature. These carriers are assumed to have a carrier frequency of f_c and a reference phase of θ. [Any variation in θ, such as would occur in practical carrier recovery circuits [2], is neglected here, and θ is assumed to be constant.]

The outputs of the two coherent demodulators are next passed through two lowpass filters B_1 and B_2. These filters remove the high frequency components which

result as a consequence of the demodulation process. Each filter has a real valued impulse response b(t) and a Fourier transform B(f), which bandlimits the signals at their inputs from -1/2T to 1/2T Hz. B(f) is illustrated in Fig 3.2.6. The outputs from the two filters B_1 and B_2 are, respectively, $r_1(t)$ and $r_2(t)$. Imaginary values are now assigned to $r_2(t)$, which is then added to $r_1(t)$, to give the resultant received signal r(t). r(t) is sampled once per signal element at the time instants $t = iT$ to give the samples $\{r_i\}$ which are then fed to the detection system.

The QAM signal at the input to the transmission path is

$$x(t) = \sqrt{2}\left\{ \sum_i s_{1,i}\, a(t-iT) \cos 2\pi f_c t - \sum_i s_{2,i}\, a(t-iT) \sin 2\pi f_c t \right\} \tag{3.2.1}$$

$$= \sqrt{2} \sum_i s_{1,i}\, a(t-iT) \left[\frac{e^{j2\pi f_c t} + e^{-j2\pi f_c t}}{2} \right]$$

$$- \sqrt{2} \sum_i s_{2,i}\, a(t-iT) \left[\frac{e^{j2\pi f_c t} - e^{-j2\pi f_c t}}{2j} \right]$$

$$= \frac{1}{\sqrt{2}} \sum_i a(t-iT) \left\{ (s_{1,i} + js_{2,i})\, e^{j2\pi f_c t} + (s_{1,i} - js_{2,i})\, e^{-j2\pi f_c t} \right\} \tag{3.2.2}$$

Let

$$s_i = s_{1,i} + js_{2,i} \tag{3.2.3}$$

where s_i is a complex data value with $j = \sqrt{-1}$. Eqn. 3.2.2 can now be written as

$$x(t) = \frac{1}{\sqrt{2}} \left\{ \sum_i \left[s_i\, e^{j2\pi f_c t} + s_i^*\, e^{-j2\pi f_c t} \right] a(t-iT) \right\} \tag{3.2.4}$$

where s_i^* denotes the complex conjugate of s_i. The signal at the input to the coherent demodulator is

$$z(t) = x(t) * h(t) * c(t) + n(t) * c(t) \qquad (3.2.5)$$

where * denotes convolution. The signals at the outputs of the two lowpass filters B_1 and B_2 are, respectively,

$$r_1(t) = \{ \sqrt{2}\ z(t) \cos(2\pi f_c t+\theta) \} * b(t) \qquad (3.2.6)$$

and

$$r_2(t) = \{ -\sqrt{2}\ z(t) \sin(2\pi f_c t+\theta) \} * b(t) \qquad (3.2.7)$$

where θ is the phase of the reference carrier relative to that of the signal carrier. The operator * in eqns. 3.2.5, 3.2.6 and 3.2.7, and indeed, in the rest of the chapter, is taken to denote the process of convolution.

The received signal at the input to the sampler is the complex signal

$$r(t) = r_1(t) + jr_2(t) \qquad (3.2.8)$$

Substituting from eqns. 3.2.6 and 3.2.7 gives

$$r(t) = \{ \sqrt{2}\ z(t) [\cos(2\pi f_c t+\theta) - j \sin(2\pi f_c t+\theta)] \} * b(t)$$

$$= \{ \sqrt{2}\ z(t)\, e^{-j(2\pi f_c t+\theta)} \} * b(t) \qquad (3.2.9)$$

Substituting the value of z(t) from eqn. 3.2.5 in eqn. 3.2.9 gives,

$$r(t) = \{ \sqrt{2}\ [x(t) * h(t) * c(t) + n(t) * c(t)]\ e^{-j(2\pi f_c t+\theta)} \} * b(t)$$

$$= \left\{ \sqrt{2}\ [\, x(t) * h(t) * c(t)\,]\ e^{-j(2\pi f_c t + \theta)} + \sqrt{2}\ [\, n(t) * c(t)\,]\ e^{-j(2\pi f_c t + \theta)} \right\} * b(t) \qquad (3.2.10)$$

Substituting the value of x(t) from eqn. 3.2.4, in eqn. 3.2.10 gives

$$r(t) = \left\{ \left[\left\{ \sum_i (s_i e^{j2\pi f_c t} + s_i^* e^{-j2\pi f_c t})\, a(t-iT) \right\} * h(t) * c(t) \right] e^{-j(2\pi f_c t + \theta)} + \sqrt{2}\ [\, n(t) * c(t)\,]\ e^{-j(2\pi f_c t + \theta)} \right\} * b(t) \qquad (3.2.11)$$

With the aid of the relationship [6]

$$\{ f_1(t) * f_2(t) \}\ e^{-j2\pi f_c t} = f_1(t)\, e^{-j2\pi f_c t} * f_2(t)\, e^{-j2\pi f_c t} \qquad (3.2.12)$$

eqn. 3.2.11 can be written as

$$r(t) = \left\{ \left[\left\{ \sum_i (s_i e^{j2\pi f_c t} + s_i^* e^{-j2\pi f_c t})\, a(t-iT) \right\} e^{-j2\pi f_c t} * \{ h(t) * c(t) \}\ e^{-j2\pi f_c t} \right] e^{-j\theta} + \sqrt{2}\ [\, n(t) * c(t)\,]\ e^{-j(2\pi f_c t + \theta)} \right\} * b(t)$$

$$= \left[\sum_i (s_i + s_i^* e^{-j4\pi f_c t})\, a(t-iT) * [\, h(t) * c(t)\,]\ e^{-j2\pi f_c t} \right] e^{-j\theta} * b(t) + \left[\sqrt{2}\ [\, n(t) * c(t)\,]\ e^{-j(2\pi f_c t + \theta)} \right] * b(t)$$

$$= \left[\sum_i s_i\, a(t-iT) * [\, h(t) * c(t)\,]\ e^{-j2\pi f_c t} \right] e^{-j\theta} * b(t) + \left[\sum_i s_i^* e^{-j4\pi f_c t}\, a(t-iT) * [\, h(t) * c(t)\,]\ e^{-j2\pi f_c t} \right] e^{-j\theta} * b(t) + \left[\sqrt{2}\ [\, n(t) * c(t)\,]\ e^{-j(2\pi f_c t + \theta)} \right] * b(t) \qquad (3.2.13)$$

The spectrum of $\exp(-j4\pi f_c t)\, a(t)$ lies outside the passband of the two lowpass filters B_1 and B_2, and therefore, the second term in eqn. 3.2.13 reduces to zero, giving,

$$r(t) = \sum_i s_i \left\{ a(t-iT) * [\, h(t) * c(t)\,] e^{-j2\pi f_c t} \right\} e^{-j\theta} * b(t)$$

$$+ \sqrt{2}\, [\, n(t) * c(t)\,] e^{-j(2\pi f_c t + \theta)} * b(t) \qquad (3.2.14)$$

$$r(t) = \sum_i s_i\, y(t-iT) + w(t) \qquad (3.2.15)$$

where,

$$y(t) = \left\{ a(t) * [\, h(t) * c(t)\,] e^{-j2\pi f_c t} \right\} e^{-j\theta} * b(t) \qquad (3.2.16)$$

and

$$w(t) = \left\{ \sqrt{2}\, [\, n(t) * c(t)\,] e^{-j(2\pi f_c t + \theta)} \right\} * b(t) \qquad (3.2.17)$$

y(t) here, is the overall system impulse response which is complex valued and w(t) is the resultant complex valued noise component in the received signal.

y(t), in eqn. 3.2.16, does not appear to be a baseband waveform due to the component $\exp(-j2\pi f_c t)$. However, consider $h(t)*c(t)$. This is a bandpass waveform whose spectrum is centred at f_c Hz. Due to the multiplying factor $\exp(-j2\pi f_c t)$, the spectrum of $h(t)*c(t)$ is shifted in frequency by $-f_c$ Hz., thus making $[h(t)*c(t)]\exp(-j2\pi f_c t)$ a lowpass response. Together with the bandlimiting applied by A(f) and B(f), this results in y(t) being converted to a baseband response. Moreover, the value of θ, and those of a(t), h(t), c(t) and b(t) are either known or can easily be determined from their corresponding frequency characteristics. Thus, the channel model in eqn. 3.2.16 could be taken to represent the baseband equivalent of a bandpass QAM system, and is shown in Fig. 3.2.7.

It is important, at this stage, to consider in greater detail, the statistics of the complex valued Gaussian noise waveform w(t). In this model, since the amplitude

spectrum |C(f)|, of the receiver bandpass filter is symmetrical about the carrier frequency f_c, it could be shown that, any sample of the real part of w(t), and any sample of the imaginary part of w(t) are statistically independent Gaussian random variables, with zero mean and a fixed variance [5,6] .

From eqns. 3.2.12 and 3.2.17, the noise function, w(t), could be expressed as

$$w(t) = \left\{ \sqrt{2} \left[n(t)\, e^{-j2\pi f_c t} * c(t)\, e^{-j2\pi f_c t} \right] e^{-j\theta} \right\} * b(t) \qquad (3.2.18)$$

Since the two-sided power spectral density of n(t) is $N_0/2$, and multiplication by $\exp(-j2\pi f_c t)$ merely shifts the spectrum of n(t) by $-f_c$ without affecting its power denstiy, it could be concluded that the power spectral density of $n(t) \exp(-j2\pi f_c t)$ is $N_0/2$. Thus, from eqn. 3.2.18,

$$|W(f)|^2 = 2 \cdot \frac{1}{2} N_0 \cdot |C(f+f_c)|^2 |B(f)|^2 \qquad (3.2.19)$$

where $|W(f)|^2$ is the power spectral density of w(t).

The autocorrelation function is the inverse discrete Fourier transform of the power spectral density. Thus, if the autocorrelation function of w(t) is $R_w(\tau)$, then, from eqn. 3.2.19,

$$R_w(\tau) = N_0 \int_{-\infty}^{\infty} |C(f+f_c)|^2 |B(f)|^2 e^{j2\pi f\tau}\, df \qquad (3.2.20)$$

If the transfer functions of the filters C, B_1 and B_2 are assumed to be as given in this model, then from eqn. 3.2.20,

$$R_w(\tau) = N_0 \int_{-1/2T}^{1/2T} e^{j2\pi f\tau}\, df$$

$$= \frac{N_0}{T} \frac{\sin(\pi\tau/T)}{\pi\tau/T} \qquad (3.2.21)$$

The variance of a random process is given by its autocorrelation function at zero lag. Thus, if the variance of w(t) is $(\sigma_w)^2$, then, from eqn. 3.2.21,

$$\sigma_w^2 = R_w(0) = \frac{N_0}{T} \qquad (3.2.22)$$

It could also be shown that the autocorrelation function of each of the real and imaginary parts of w(t) is given by half the real part of the autocorrelation function of w(t) [5,6,7]. In the case considered here, since |C(f)| is symmetric about f_c over positive frequencies, the autocorrelation function given in eqn. 3.2.20 is purely real. Thus the autocorrelation function $R_w'(\tau)$, of each of the real and imaginary parts of w(t) is given by,

$$R_w'(\tau) = \frac{1}{2} N_0 \int_{-\infty}^{\infty} |C(f+f_c)|^2 |B(f)|^2 e^{j2\pi f\tau} \, df \qquad (3.2.23)$$

For the model considered here, this leads to a variance, $(\sigma_w')^2$, for each of the real and imaginary parts of w(t), of

$$(\sigma_w')^2 = \frac{N_0}{2T} \qquad (3.2.24)$$

The transmitted and received signal energies are two other quantities that are particularly important in assessing the performance of a system, especially since they are needed in the calculation of signal-to-noise ratios. Let the average transmitted energy per signal element be E_{TS}. Then,

$$E_{TS} = E\left[|s_i|^2 \int_{-\infty}^{\infty} a^2(t) \, dt \right] \qquad (3.2.25)$$

where E[.] denotes the expected value. Since the $\{s_i\}$ are statistically independent and have zero mean, eqn. 3.2.25 could be written as,

$$E_{TS} = \overline{s_i^2} \int_{-\infty}^{\infty} a^2(t)\, dt \qquad (3.2.26)$$

where the first term on the right hand side of the equation is the expected value of $|s_i|^2$. Using Parseval's theorem, and assuming the given transmitter filters, eqn. 3.2.26 could be written as

$$E_{TS} = \overline{s_i^2} \int_{-1/2T}^{1/2T} |A(f)|^2\, df \qquad (3.2.27)$$

If the average transmitted energy per bit is E_{TB}, then,

$$E_{TB} = \frac{\overline{s_i^2}}{\log_2 m} \int_{-1/2T}^{1/2T} |A(f)|^2\, df \qquad (3.2.28)$$

where m is the level of the QAM signal. In the case considered here, since m = 4,

$$E_{TB} = \frac{\overline{s_i^2}}{2} \int_{-1/2T}^{1/2T} |A(f)|^2\, df \qquad (3.2.29)$$

The average received energy per signal element is given by

$$E_{RS} = E\left[|s_i|^2 \int_{-\infty}^{\infty} y^2(t)\, dt \right] \qquad (3.2.30)$$

Using the fact that the $\{s_i\}$ are statistically independent and have zero mean, and employing Parseval's theorem, eqn.3.2.30 could be written as

$$E_{RS} = \overline{s_i^2} \int_{-\infty}^{\infty} |Y(f)|^2\, df \qquad (3.2.31)$$

where Y(f) is the transfer function of y(t). From eqn. 3.2.16 it could be seen that ,

$$|Y(f)|^2 = |A(f)|^2 |H(f+f_c)|^2 |C(f+f_c)|^2 |B(f)|^2 \quad (3.2.32)$$

Substituting the value of eqn. 3.2.32 in eqn. 3.2.31 gives

$$E_{RS} = \overline{s_i^2} \int_{-\infty}^{\infty} |A(f)|^2 |H(f+f_c)|^2 |C(f+f_c)|^2 |B(f)|^2 \, df \quad (3.2.33)$$

Assuming the filters C, B_1, and B_2 used in this model, eqn. 3.2.33 could be written as

$$E_{RS} = \overline{s_i^2} \int_{-1/2T}^{1/2T} |A(f)|^2 |H(f+f_c)|^2 \, df \quad (3.2.34)$$

If the average received signal energy per bit is E_{RB}, then,

$$E_{RB} = \frac{\overline{s_i^2}}{\log_2 m} \int_{-1/2T}^{1/2T} |A(f)|^2 |H(f+f_c)|^2 \, df \quad (3.2.35)$$

where m is as defined previously.

3.3 QAM SYSTEM MODEL WHEN USED OVER HF RADIO LINKS

It is now appropriate to discuss the model of a data transmission system employing QAM over HF radio links. An illustration of such a model is given in Fig. 3.3.1. This is an example of a synchronous, serial data transmission system, where the transmitted signal is subjected to Rayleigh fading, as would be the case in a practical HF radio channel.

The $\{s_i\}$ are a sequence of complex data symbol values, represented as a series of impulses, where s_i could assume the value of any one of a possible set of complex numbers, and 1/T is the signal element rate of the system in bauds. The $\{s_i\}$ here, carry the information to be transmitted, and are statistically independent and equally likely to

have any one of their possible values. The filter A' is a lowpass filter which shapes the spectrum of the transmitted signal such that the resultant QAM signal fits into the passband of the voice frequency channel in the HF radio spectrum. The impulse response of A' is the real valued function a'(t), and its transfer function is A'(f), which is assumed to be such that,

$$A'(f) = 0 \quad \text{for} \quad f < -f_c + kf_{sp} \quad \text{and} \quad f > f_c - kf_{sp} \tag{3.3.1}$$

where f_c is the carrier frequency in Hz. f_{sp} here, is the largest value of the frequency spread (in Hz.) that is expected to be introduced by the HF channel and k is an appropriate positive integer [6]. The reason for this assumption will become clear later on, but, at this stage, suffice it to say that the above assumption facilitates effective lowpass filtering at the receiver. A possible example of the absolute value of A'(f), i.e., |A'(f)|, is shown in Fig. 3.3.2. The output of the lowpass filter A' is multiplied by the complex factor $\sqrt{2}\exp(j2\pi f_c t)$ to give the signal $x_1(t)$. The real part of $x_1(t)$ is the QAM signal (as given in Section 3.2), and is denoted by $x_2(t)$. A possible example of the absolute value of the transfer function of $x_2(t)$, i.e. $|X_2(f)|$, is shown in Fig. 3.3.3.

The spectrum of the voiceband QAM signal is next shifted to the HF radio spectrum, (3-30 MHz.), by a process of single sideband surpressed carrier amplitude modulation [1]. The radio transmitter filter employed towards this end is G, which has an impulse response g(t) and a transfer function G(f). The modulated HF radio signal x(t), is transmitted over an HF radio channel, whose model is as explained in Section 2.6, and as shown in Fig. 3.3.4. For simplicity's sake, the two skywave model was chosen for this purpose, where the relative delay between the skywaves is τ seconds. The Rayleigh faded and distorted signal at the output of the HF radio channel is denoted by z(t), and is now linearly demodulated by the radio receiver, in order to shift its spectrum back to the voice frequency band. The radio receiver filter employed in the demodulation is D, which has an impulse response d(t) and a transfer function D(f). White Gaussian noise, (denoted by n(t)), with zero mean and a two-sided power spectral density of $N_0/2$ is added to the signal at the output of the radio receiver filter D. It should be noted here, that the processes of modulation and demodulation are linear, and they do not change the bandwidth of the signal, or introduce any distortion into the signal, other than that introduced by the HF radio channel and the radio equipment filters themselves. Thus, these processes are not explicitly shown in Fig .3.3.1, rather, the entire HF radio link is represented by its equivalent model in the voiceband, as shown in the figure.

The noisy, Rayleigh faded, and distorted QAM signal is next passed through a bandpass filter C, whose impulse response and transfer function are, respectively, c(t) and C(f). This filter is found in the data modem itself, and its function is to remove the noise frequencies which lie outside the bandwidth of the signal, without unduly distorting it. |C(f)| is assumed to be symmetric about f_c over positive frequencies. The signal at the output of the filter C, undergoes a process of coherent demodulation, where it is first multiplied by the complex valued reference signal $\sqrt{2}\exp(j2\pi f_c't)$. f_c' is equal to the average instantaneous frequency of the received signal, and, in the case considered here, is assumed to be equal to the carrier frequency f_c, thus eliminating any frequency offset that may have been introduced by the channel. This, in itself, is a fair assumption, especially for the theoretical analysis of the system, since in any practical modem, the carrier recovery circuit is expected to remove the above mentioned frequency offset by, for example, employing the well known phase locked loop technique [1,2] .

The signal at the output of the complex multiplier is now subjected to the second stage of the demodulation process, which consists of it being passed through a lowpass filter B', with impulse response b'(t) and transfer function B'(f), where the value of B'(f) is zero outside the frequency band $-f_c$ to f_c . The function of this lowpass filter is to surpress the harmonics generated by the multiplying process, thus leaving only the baseband received signal, r(t), at its output. This received signal is sampled once per signal element, and passed onto the detector, whose output gives the desired, detected signal s_i' .

From Fig. 3.3.1, the signal at the output of the first multiplier is given by

$$x_1(t) = \sqrt{2} \sum_i s_i\, a'(t-iT)\, e^{j2\pi f_c t} \qquad (3.3.2)$$

Taking the real part of $x_1(t)$ gives,

$$x_2(t) = \sqrt{2}\ \mathrm{Re}\left\{ \sum_i s_i\, a'(t-iT)\, e^{j2\pi f_c t} \right\} \qquad (3.3.3)$$

Let the complex valued data symbol, s_i , be given by

$$s_i = s_{1,i} + j s_{2,i} \qquad (3.3.4)$$

Then from eqn. 3.3.2,

$$x_2(t) = \sqrt{2}\ \mathrm{Re}\left\{ \sum_i (s_{1,i} + j s_{2,i})\, a'(t-iT)\, e^{j2\pi f_c t} \right\}$$

$$= \sqrt{2}\ \mathrm{Re}\left\{ \sum_i (s_{1,i} + j s_{2,i})\, a'(t-iT)\, [\cos 2\pi f_c t + j \sin 2\pi f_c t] \right\}$$

$$= \sqrt{2}\left\{ \sum_i s_{1,i}\, a'(t-iT) \cos 2\pi f_c t - \sum_i s_{2,i}\, a'(t-iT) \sin 2\pi f_c t \right\} \qquad (3.3.5)$$

Eqn. 3.3.5. is exactly similar to eqn. 3.2.1 in the previous section. Thus, from eqn. 3.2.1, 3.2.4 and 3.3.5, the QAM signal $x_2(t)$ could be written as

$$x_2(t) = \frac{1}{\sqrt{2}} \left\{ s_i\, a'(t-iT)\, e^{j2\pi f_c t} + s_i^*\, a'(t-iT)\, e^{-j2\pi f_c t} \right\} \qquad (3.3.6)$$

The signal at the input to the transmission path is given by x(t) where

$$x(t) = x_2(t) * g(t)$$

$$= \frac{1}{\sqrt{2}} \sum_i \left\{ s_i\, a'(t-iT)\, e^{j2\pi f_c t} + s_i^*\, a'(t-iT)\, e^{-j2\pi f_c t} \right\} * g(t)$$

$$= \frac{1}{\sqrt{2}} \sum_i \left\{ s_i\, a'(t-iT)\, e^{j2\pi f_c t} * g(t) + s_i^*\, a'(t-iT)\, e^{-j2\pi f_c t} * g(t) \right\}$$

$$= \frac{1}{\sqrt{2}} \sum_i \left\{ [\, s_i\, a'(t-iT) * g(t)\, e^{-j2\pi f_c t}\,]\, e^{j 2\pi f_c t} \right.$$

$$\left. + [\, s_i^*\, a'(t-iT) * g(t)\, e^{j2\pi f_c t}\,]\, e^{-j 2\pi f_c t} \right\} \qquad (3.3.7)$$

Let

$$a(t\text{-}iT) = a'(t\text{-}iT) * g(t)\, e^{-j2\pi f_c t} \qquad (3.3.8)$$

From eqns. 3.3.7 and 3.3.8, x(t) can be written as

$$x(t) = \frac{1}{\sqrt{2}} \left\{ \sum_i s_i\, a(t\text{-}iT)\, e^{j2\pi f_c t} + \sum_i s_i^*\, a^*(t\text{-}iT)\, e^{-j2\pi f_c t} \right\} \qquad (3.3.9)$$

a(t-iT) here represents the overall filtering carried out on the transmitter side, on the baseband signal.

x(t) is the HF radio signal that is assumed to be transmitted over the HF radio channel. Since the channel model involves Hilbert transforms, (see Fig. 3.3.4), it is first necessary to define such. A Hilbert transformer is an allpass filter which introduces a -90°/+90° phase shift to all positive/negative frequencies of its input signal [2]. It is characterized by the impulse response

$$p(t) = \frac{1}{\pi t} \qquad (3.3.10)$$

and the transfer function

$$P(f) = \begin{cases} -j & \text{for } f > 0 \\ +j & \text{for } f < 0 \end{cases} \qquad (3.3.11)$$

If the Hilbert transform of x(t) is denoted by $x_H(t)$, then

$$x_H(t) = x(t) * p(t) \qquad (3.3.12)$$

From eqn. 3.3.9, substituition for x(t) in eqn. 3.3.12 yields

$$x_H(t) = \frac{1}{\sqrt{2}} \left\{ \sum_i s_i\, a(t\text{-}iT)\, e^{j2\pi f_c t} * p(t) + \sum_i s_i^*\, a^*(t\text{-}iT)\, e^{-j2\pi f_c t} * p(t) \right\} \qquad (3.3.13)$$

Noting the relationship in eqn. 3.2.12 in the previous section, eqn. 3.3.13 could be written as

$$x_H(t) = \frac{1}{\sqrt{2}} \{ [\sum_i s_i\, a(t-iT) * p(t)\, e^{-j2\pi f_c t}]\, e^{j2\pi f_c t} + [\sum_i s_i^*\, a^*(t-iT) * p(t)\, e^{j2\pi f_c t}]\, e^{-j2\pi f_c t} \} \qquad (3.3.14)$$

From eqns. 3.3.8 and 3.3.1, it follows that the Fourier transform of a(t) is bandlimited to the frequency band of A'(f), i.e., $-f_c$ to f_c. Thus, if a Fourier transform were to be taken of $x_H(t)$ in eqn. 3.3.14, the frequency band of interest would be $-f_c$ to f_c. Now, the Fourier transforms of $p(t)\exp(-j2\pi f_c t)$ and $p(t)\exp(j2\pi f_c t)$ are given, respectively, by $P(f+f_c)$ and $P(f-f_c)$, where P(f) is as given in eqn. 3.3.11. The values of $P(f+f_c)$ and $P(f-f_c)$ over the frequency band of interest (namely, $-f_c$ to f_c) are, respectively, -j and +j. Thus, by taking the Fourier transform of $x_H(t)$ in eqn. 3.3.14, inserting the appropriate values of $P(f+f_c)$, and $P(f-f_c)$, and then taking the inverse Fourier transform of the resultant function, $x_H(t)$ could be rewritten as

$$x_H(t) = \frac{1}{\sqrt{2}} \{ \sum_i s_i\, a(t-iT)\,(-j)\, e^{j2\pi f_c t} + \sum_i s_i^*\, a^*(t-iT)\,(j)\, e^{-j2\pi f_c t} \} \qquad (3.3.15)$$

Referring to Figs. 3.3.1 and 3.3.4, it could be seen that the signal at the output of the HF channel is given by

$$z(t) = x(t)\, q_1(t) + x_H(t)\, q_2(t) + x(t-\tau)\, q_3(t) + x_H(t-\tau)\, q_4(t) \qquad (3.3.16)$$

Substituting the values of eqns. 3.3.9 and 3.3.15 in eqn. 3.3.16 gives

$$z(t) = \frac{1}{\sqrt{2}} \sum_i \{ [s_i\, a(t-iT)\, e^{j2\pi f_c t} + s_i^*\, a^*(t-iT)\, e^{-j2\pi f_c t}]\, q_1(t) + [(-j)\, s_i\, a(t-iT)\, e^{j2\pi f_c t} + j\, s_i^*\, a^*(t-iT)\, e^{-j2\pi f_c t}]\, q_2(t)$$

$$+ \left[s_i\, a(t-\tau-iT)\, e^{j2\pi f_c(t-\tau)} + s_i^{*}\, a^{*}(t-\tau-iT)\, e^{-j2\pi f_c(t-\tau)} \right] q_3(t)$$

$$+ \left[(-j)\, s_i\, a(t-\tau-iT)\, e^{j2\pi f_c(t-\tau)} + j\, s_i^{*}\, a^{*}(t-\tau-iT)\, e^{-j2\pi f_c(t-\tau)} \right] q_4(t) \Big\}$$

$$= \frac{1}{\sqrt{2}} \sum_i \Big\{ s_i\, a(t-iT)\, [q_1(t) - jq_2(t)]\, e^{j2\pi f_c t}$$

$$+ s_i^{*}\, a^{*}(t-iT)\, [q_1(t) + jq_2(t)]\, e^{-j2\pi f_c t}$$

$$+ s_i\, a(t-\tau-iT)\, [q_3(t) - jq_4(t)]\, e^{j2\pi f_c(t-\tau)}$$

$$+ s_i^{*}\, a^{*}(t-\tau-iT)\, [q_3(t) + jq_4(t)]\, e^{-j2\pi f_c(t-\tau)} \qquad (3.3.17)$$

Let,

$$h_i(t-iT) = a(t-iT)\,[q_1(t) - jq_2(t)] + \{ a(t-\tau-iT)\,[q_3(t) - jq_4(t)] \}\, e^{-j2\pi f_c \tau} \qquad (3.3.18)$$

Using the value in eqn. 3.3.18, eqn. 3.3.17 could be rewritten as

$$z(t) = \frac{1}{\sqrt{2}} \sum_i \left\{ s_i\, h_i(t-iT)\, e^{j2\pi f_c t} + s_i^{*}\, h_i^{*}(t-iT)\, e^{-j2\pi f_c t} \right\} \qquad (3.3.19)$$

Consider eqn. 3.3.18. In the case considered here, the time delay τ is assumed to be a constant, and hence the term $\exp(-j2\pi f_c \tau)$ is merely a complex valued scalar with an absolute value of unity. Therefore, it has no bearing on the statistical properties of $[q_3(t)-jq_4(t)]\exp(-j2\pi f_c \tau)$, especially since $q_3(t)$ and $q_4(t)$ are statistically independent random variables with zero mean. Moreover, it does not change the power spectrum of the above mentioned signal. It could thus be ignored in eqn. 3.3.18, enabling $h_i(t-iT)$ to be rewritten as

$$h_i(t\text{-}iT) = a(t\text{-}iT)\,[\,q_1(t) - jq_2(t)\,] + a(t\text{-}\tau\text{-}iT)\,[\,q_3(t) - jq_4(t)\,] \qquad (3.3.20)$$

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

$$r_1(t) = z(t) * d(t) * c(t) + n(t) * c(t) \qquad (3.3.21)$$

and the demodulated baseband signal at the output of the lowpass filter B', is given by

$$r(t) = \sqrt{2}\; r_1(t)\, e^{-j2\pi f_c' t} * b'(t) \qquad (3.3.22)$$

However, as mentioned previously, f_c' is equal to the carrier frequency f_c. Thus, from eqn. 3.3.22,

$$r(t) = \sqrt{2}\; r_1(t)\, e^{-j2\pi f_c t} * b'(t) \qquad (3.3.23)$$

Substituting the value of $r_1(t)$ from eqn. 3.3.21 yields,

$$r(t) = \sqrt{2}\; \{\, [\, z(t) * d(t) * c(t)\,]\, e^{-j2\pi f_c t} + [\, n(t) * c(t)\,]\, e^{-j2\pi f_c t} \,\} * b'(t)$$

$$= \sqrt{2}\; \{\, z(t)\, e^{-j2\pi f_c t} * [\, d(t) * c(t)\,]\, e^{-j2\pi f_c t} \,\} * b'(t)$$

$$+ \sqrt{2}\; \{\, [\, n(t) * c(t)\,]\, e^{-j2\pi f_c t} \,\} * b'(t) \qquad (3.3.24)$$

Now, let

$$b(t) = [\, d(t) * c(t)\,]\, e^{-j2\pi f_c' t} * b'(t) \qquad (3.3.25)$$

where b(t) represents the overall filtering carried out at the receiver side of the modem. From eqns. 3.3.24 and 3.3.25,

$$r(t) = \sqrt{2}\left\{ z(t)\, e^{-j2\pi f_c t} \right\} * b(t) + \sqrt{2}\left\{ [\, n(t) * c(t)\,]\, e^{-j2\pi f_c t} \right\} * b'(t) \quad (3.3.26)$$

Let

$$w(t) = \sqrt{2}\left\{ [\, n(t) * c(t)\,]\, e^{-j2\pi f_c t} \right\} * b'(t) \quad (3.3.27)$$

where w(t) represents the resultant noise component in the received signal r(t). Eqn. 3.3.26 could now be written as

$$r(t) = \sqrt{2}\left[z(t)\, e^{-j2\pi f_c t} \right] * b(t) + w(t) \quad (3.3.28)$$

Substituting for z(t) from eqn. 3.3.19 gives,

$$r(t) = \sum_i \left\{ s_i\, h_j(t-iT) + s_i^*\, h_i^*(t-iT)\, e^{-j4\pi f_c t} \right\} * b(t) + w(t) \quad (3.3.29)$$

Consider now, the lowpass filter with the impulse response b(t). From eqn. 3.3.25 it can be seen that the frequency response of this filter is governed by B'(f). However, it was assumed earlier that B'(f) is zero outside the frequency band $-f_c$ to f_c. Thus, the above mentioned lowpass filter will surpress all frequency components which lie outside this band.

Consider next, the response $h_i(t-iT)$. If the frequency response of this could be constrained to be within the frequency band $-f_c$ to f_c, then the second term in eqn. 3.3.29 would be reduced to zero, due to the lowpass filtering. From eqn. 3.3.20, it can be seen that the frequency response of $h_i(t-iT)$ is dependent on A'(f), and the frequency spreads introduced by the $q_i(t)$'s. If now, A'(f) could be made such that after accommodating the highest frequency spread, it still lies within the frequency band $-f_c$ to f_c, then, the frequency response of $h_i(t-iT)$ too would be limited to this band. This is the reason for A'(f) to be defined as given in eqn. 3.3.1 and shown in Fig. 3.3.2. As regards the integer k, (eqn. 3.3.1), a value of 5 is deemed more than sufficient, in order to ensure satisfactory filtering.

Eqn. 3.3.29 could now be written as

$$r(t) = \sum_i s_i h_i(t-iT) * b(t) + w(t) \qquad (3.3.30)$$

Let

$$y_i(t-iT) = h_i(t-iT) * b(t) \qquad (3.3.31)$$

Then, from eqn. 3.3.30,

$$r(t) = \sum_i s_i y_i(t-iT) + w(t) \qquad (3.3.32)$$

From eqn. 3.3.31 and 3.3.20, $y_i(t-iT)$ is ascertained to be

$$y_i(t-iT) = \{ a(t-iT)[q_1(t) - jq_2(t)] + a(t-\tau-iT)[q_3(t) - jq_4(t)] \} * b(t) \qquad (3.3.33)$$

$y_i(t-iT)$ here, is the time varying impulse response of the linear baseband channel (without the noise), as shown in Fig. 3.3.1, and the equivalent baseband model of the QAM data transmission system over HF radio links, is given by eqn. 3.3.32. A diagrammatic exposition of this model is given in Fig. 3.3.5.

It is now appropriate to explore the statistics of the resultant noise function, w(t), in more detail. From eqn. 3.3.27,

$$w(t) = \sqrt{2} \{ [n(t) * c(t)] e^{-j2\pi f_c t} \} * b'(t) \qquad (3.3.34)$$

By a similar argument to that developed in Section 3.2, the power spectral density and autocorrelation function of w(t), denoted, respectively, by $|W(f)|^2$ and $R_w(\tau')$, could be said to be given by

$$|W(f)|^2 = N_0 |C(f+f_c)|^2 |B'(f)|^2 \qquad (3.3.35)$$

and,

$$R_w(\tau') = N_0 \int_{-f_c}^{f_c} |C(f+f_c)|^2 |B'(f)|^2 e^{j2\pi f\tau'} df \qquad (3.3.36)$$

The limits of integration here, are governed by the bandwidth of the filter B'. The variance of w(t), which is $(\sigma_w)^2$, is given by

$$\sigma_w^2 = N_0 \int_{-f_c}^{f_c} |C(f+f_c)|^2 |B'(f)|^2 df \qquad (3.3.37)$$

In the case considered here, where C(f) is symmetric about the carrier frequency f_c (for positive frequencies), it could be shown that the autocorrelation function of each of the real and imaginary parts of w(t) is equal to half the autocorrelation function of w(t), and, that the real and imaginary parts of w(t) are uncorrelated [5,6,7].

The average transmitted energy per signal element at the output of the transmitter filter with impulse response a(t), is given by,

$$E_{TS} = E\left[\int_{-\infty}^{\infty} |s_i\, a(t-iT)|^2 dt \right] \qquad (3.3.38)$$

where E[.] denotes the expected value. Since the $\{s_i\}$ are statistically independent and have zero mean, eqn. 3.3.38 could be written (using Parseval's theorem) as,

$$E_{TS} = \overline{s_i^2} \int_{-\infty}^{\infty} |A(f)|^2 df \qquad (3.3.39)$$

where the first term on the right hand side of the equation denotes the expected value of $(s_i)^2$. The average transmitted energy per bit at the output of the transmitter filter with impulse response a(t), is given by,

$$E_{IB} = \frac{\overline{s_i^2}}{\log_2 m} \int_{-\infty}^{\infty} |A(f)|^2 \, df \qquad (3.3.40)$$

The average energy per signal element at the input to the receiver filter whose impulse response is b(t), is given by,

$$E_{RS} = E\left[\int_{-\infty}^{\infty} | s_i \{ a(t-iT)[q_1(t) - jq_2(t)] + a(t-\tau-iT)[q_3(t) - jq_4(t)] \} |^2 \, dt \right] \qquad (3.3.41)$$

Since the $q_i(t)$ are independent, eqn. 3.3.41 simplifies to

$$E_{RS} = \overline{s_i^2} \left[\overline{q_1^2(t)} + \overline{q_2^2(t)} + \overline{q_3^2(t)} + \overline{q_4^2(t)} \right] \int_{-\infty}^{\infty} |A(f)|^2 \, df \qquad (3.3.42)$$

and the corresponding energy per bit is,

$$E_{RB} = \frac{\overline{s_i^2}}{\log_2 m} \left[\overline{q_1^2(t)} + \overline{q_2^2(t)} + \overline{q_3^2(t)} + \overline{q_4^2(t)} \right] \int_{-\infty}^{\infty} |A(f)|^2 \, df \qquad (3.3.43)$$

where the respective components inside the square brackets are the mean square values of the functions $q_1(t)$, $q_2(t)$, $q_3(t)$ and $q_4(t)$. Since all these functions have zero mean, the above mean square values are also their variances. From eqns 3.3.39 and 3.3.43, it could be seen that,

$$E_{RS} = \left[\overline{q_1^2(t)} + \overline{q_2^2(t)} + \overline{q_3^2(t)} + \overline{q_4^2(t)} \right] E_{TS} \qquad (3.3.44)$$

Thus, if the sum of the variances of the $q_i(t)$ are equal to unity, there would be no change in the signal energy due to transmission over the HF channel. As explained in Section 2.6.2 in the previous chapter, this is the case that is assumed.

References - Chapter 3

1. **Schwartz,M.,** 'Information transmission, modulation, and noise', McGraw Hill, 1981.

2. **Proakis, J.G.,** 'Digital communications', McGraw Hill, 1983.

3. **Clark, A.P.,** 'Equalizers for digital modems', Pentech Press, 1985.

4. **Kurzweil, J. and Bradley, S.D.,** 'Modelling passband distortion in QAM data transmission systems', 23 rd Midwest symposium on Circuits and Systems, University of Toledo, pp. 114-118, August 1980.

5. **Clark, A.P.,** 'Modelling of digital communication systems', internal report, Loughborough University of Technology.

6. **Najdi, H.,** 'Digital data transmission over voice channels', Ph.D. thesis, Loughborough University of Technology, 1982.

7. **Wozencraft, J.M., and Jacobs, I.M.,** 'Principles of communication engineering', Wiley, 1965.

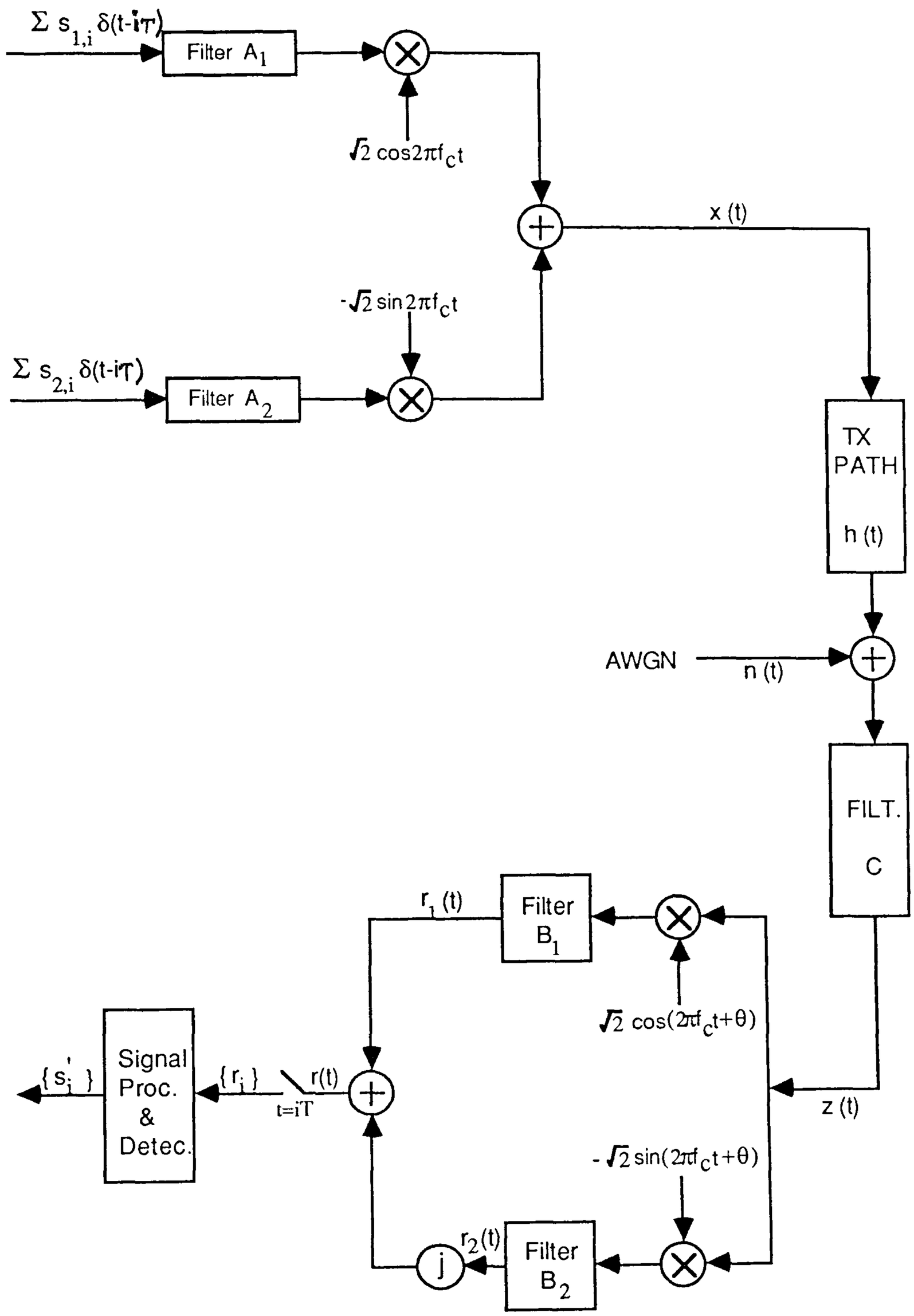


Fig. 3.2.1 Model of QAM data transmission system

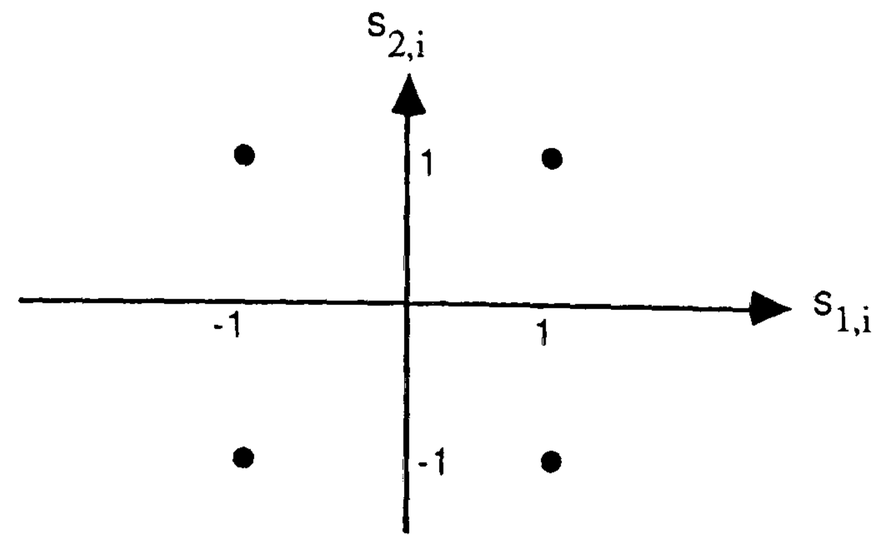


Fig. 3.2.2 A 4-level QAM signal constellation

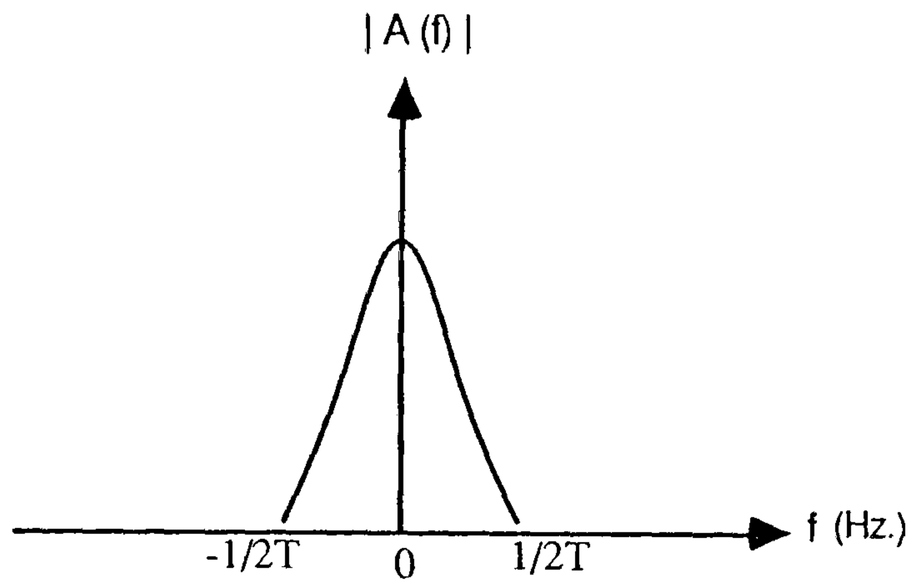


Fig. 3.2.3 Amplitude spectrum of transmitter filters A_1 and A_2

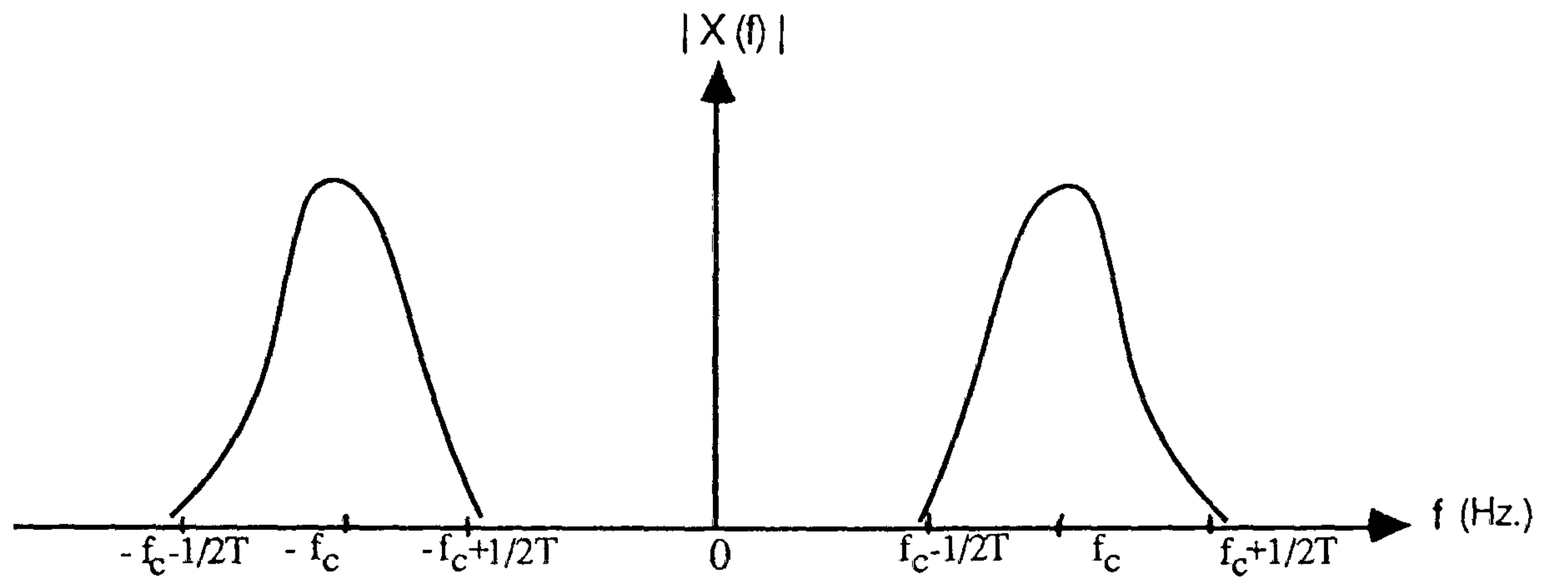


Fig. 3.2.4 Amplitude spectrum of QAM signal x(t)

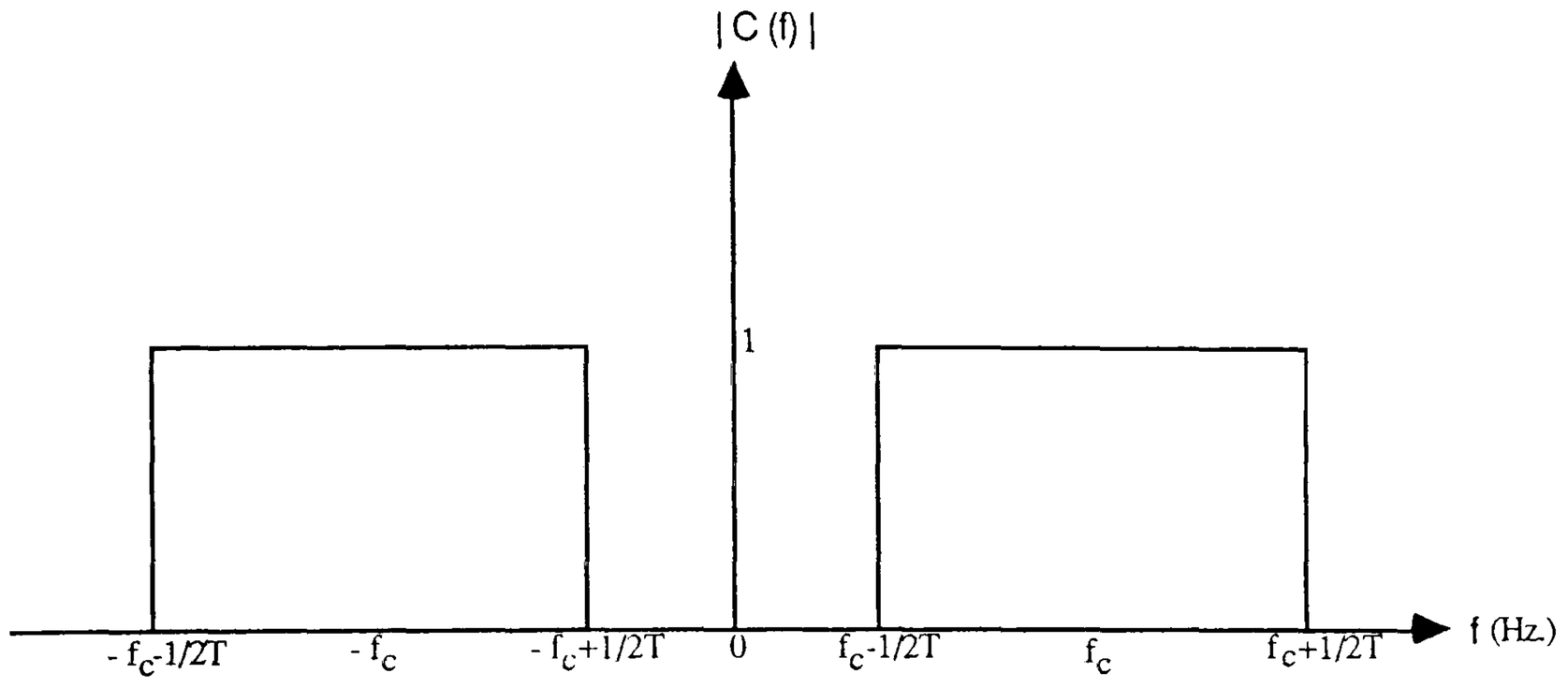


Fig. 3.2.5 Amplitude spectrum of receiver filter C

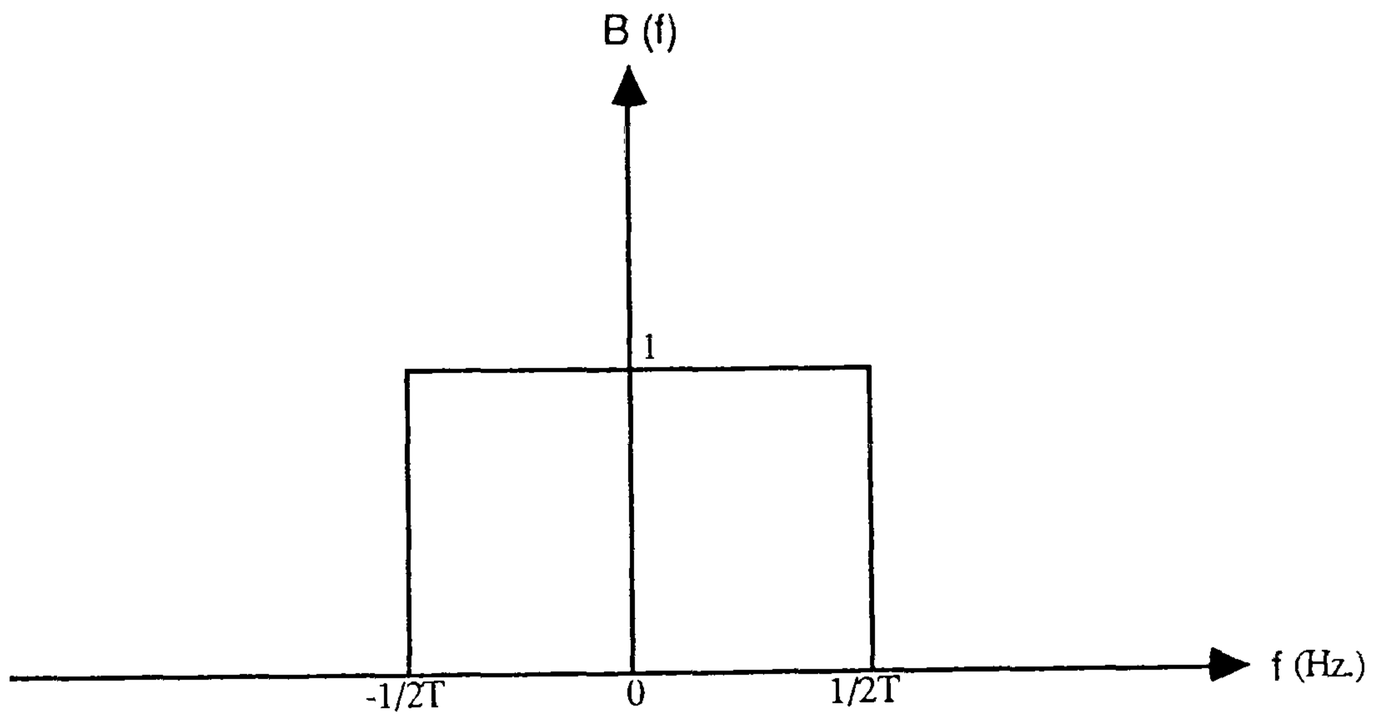
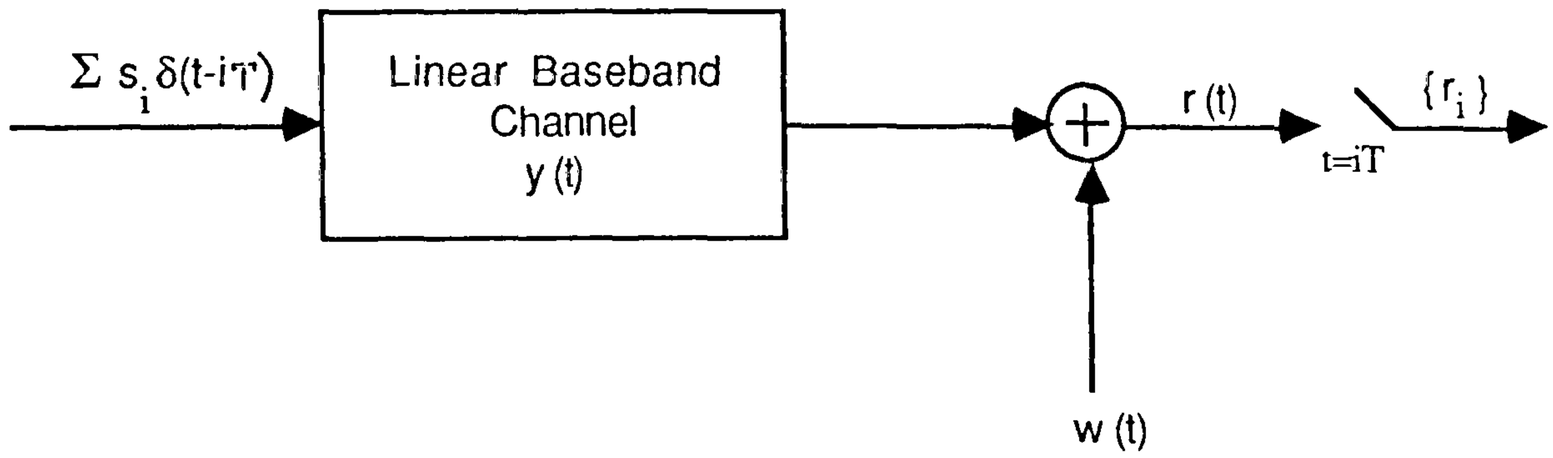


Fig. 3.2.6 Transfer function of lowpass filters B_1 and B_2



$\Sigma\, s_i\, \delta(t-iT)$ = complex valued data symbols

w (t) = complex valued additive noise as given in eqn. 3.2.17

r (t) = resultant received signal

$\{r_i\}$ = sequence of sampled values of r (t) at every iT seconds

Fig. 3.2.7 **Baseband model of bandpass QAM system**

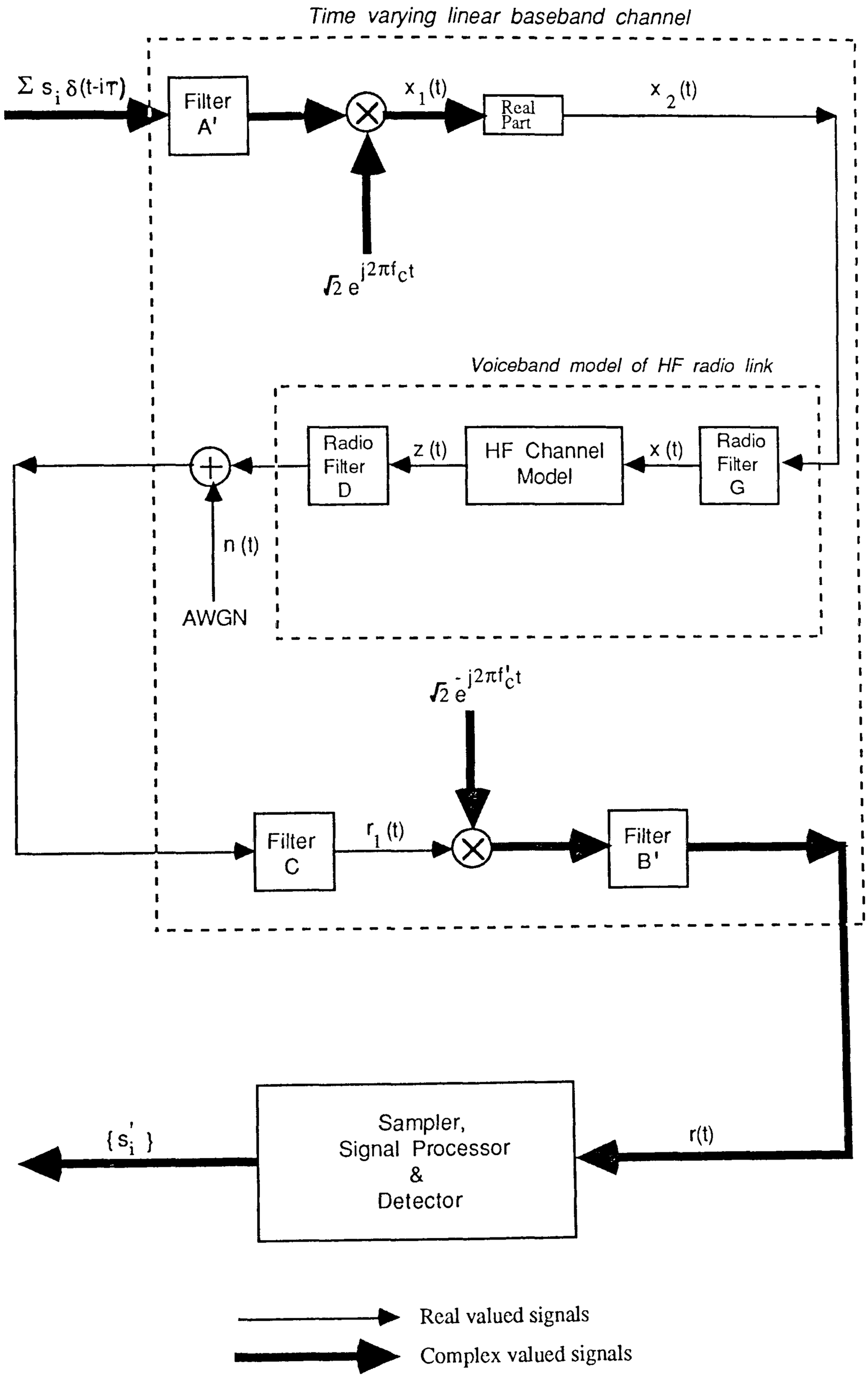


Fig. 3.3.1 QAM data transmission model over HF radio links

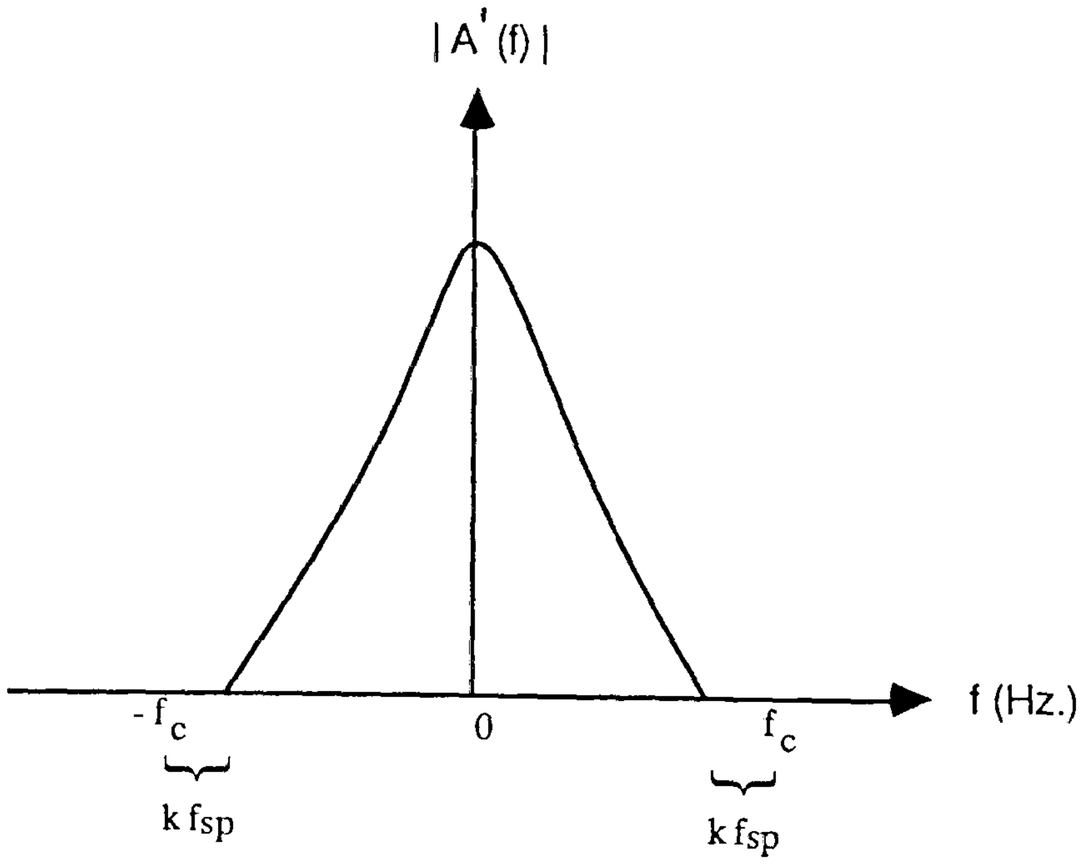


Fig. 3.3.2 Modulus of the transfer function of A'

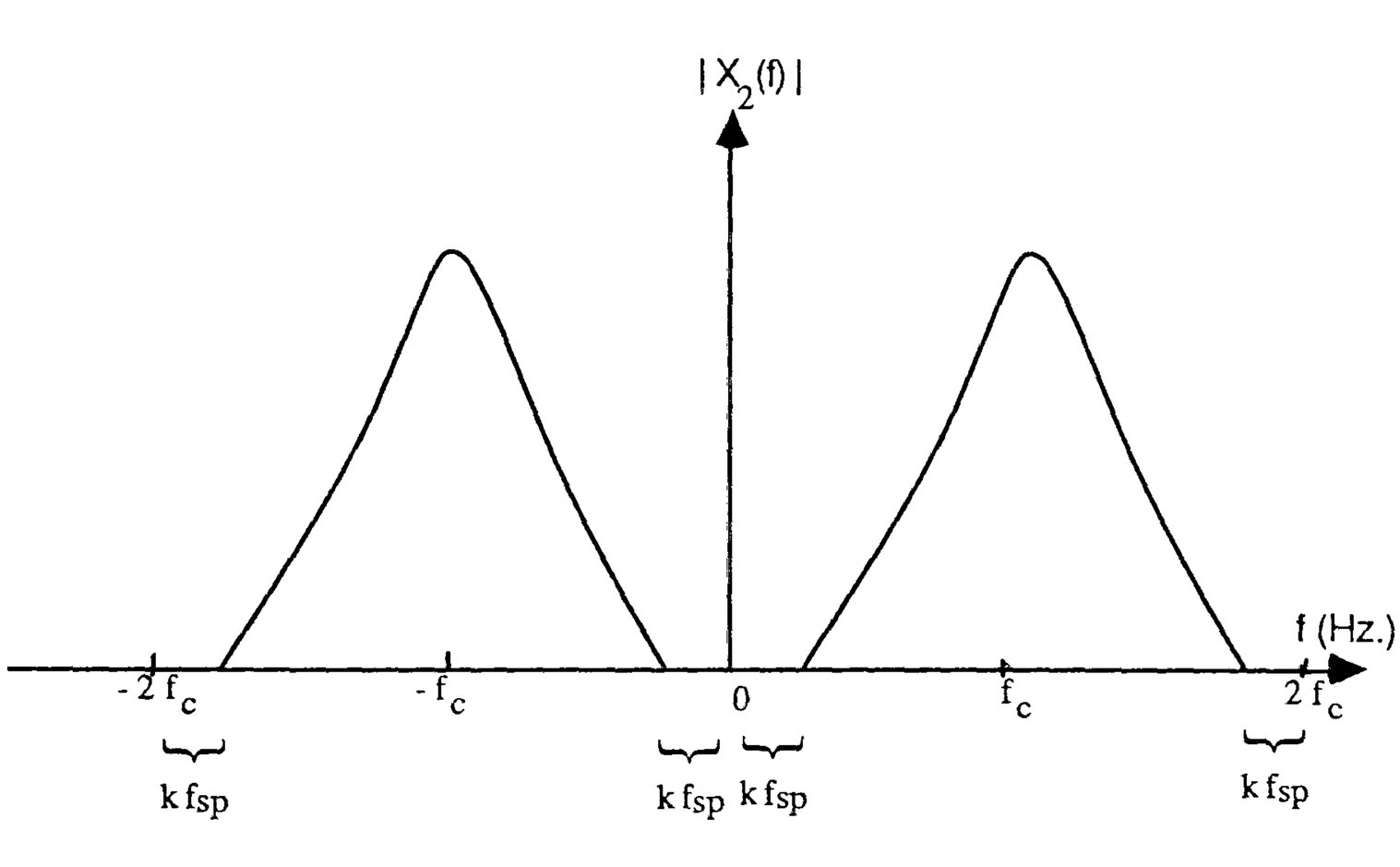


Fig. 3.3.3 Modulus of the transfer function of the QAM signal $x_2(t)$

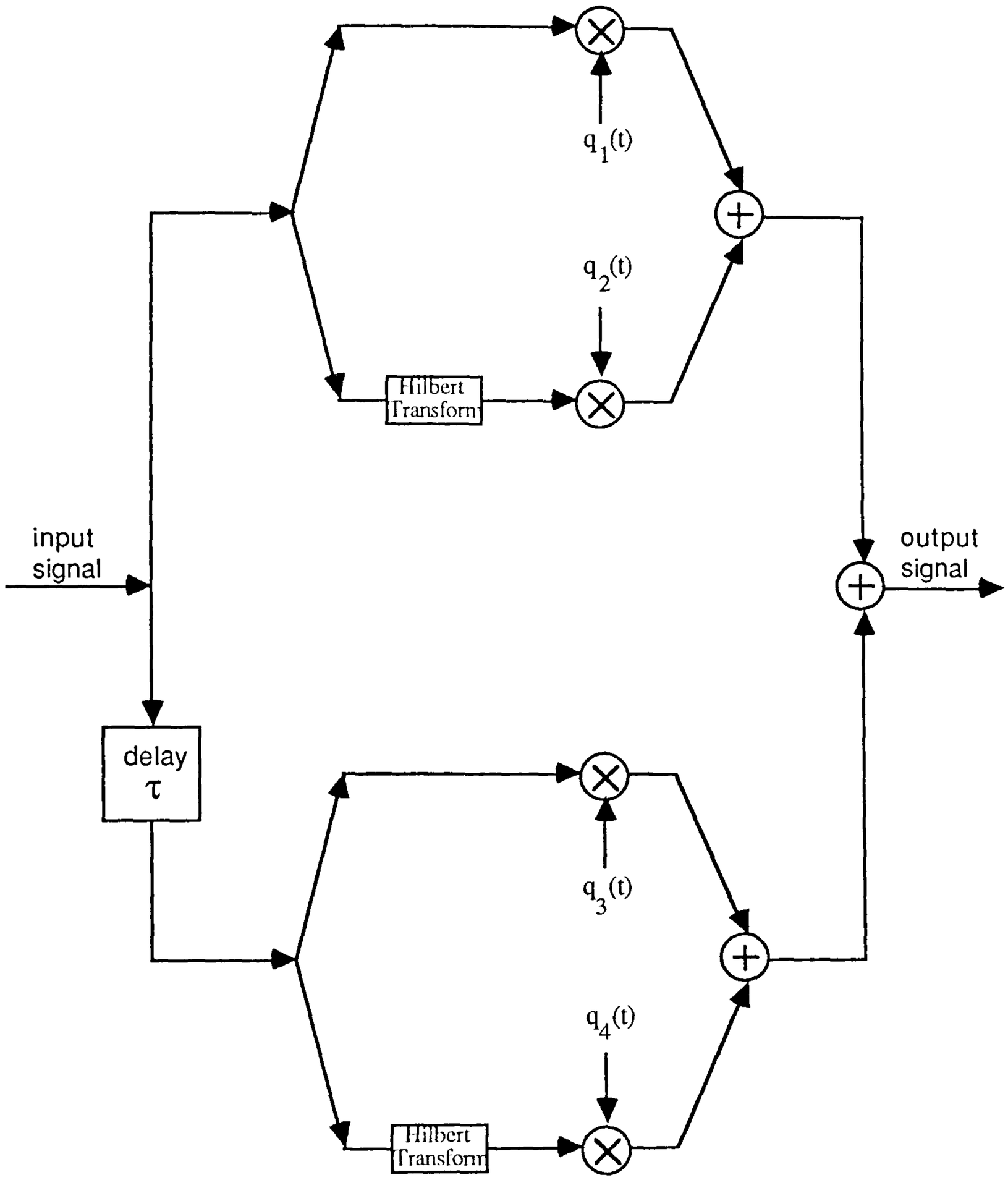


Fig. 3.3.4 HF channel model with 2 sky waves

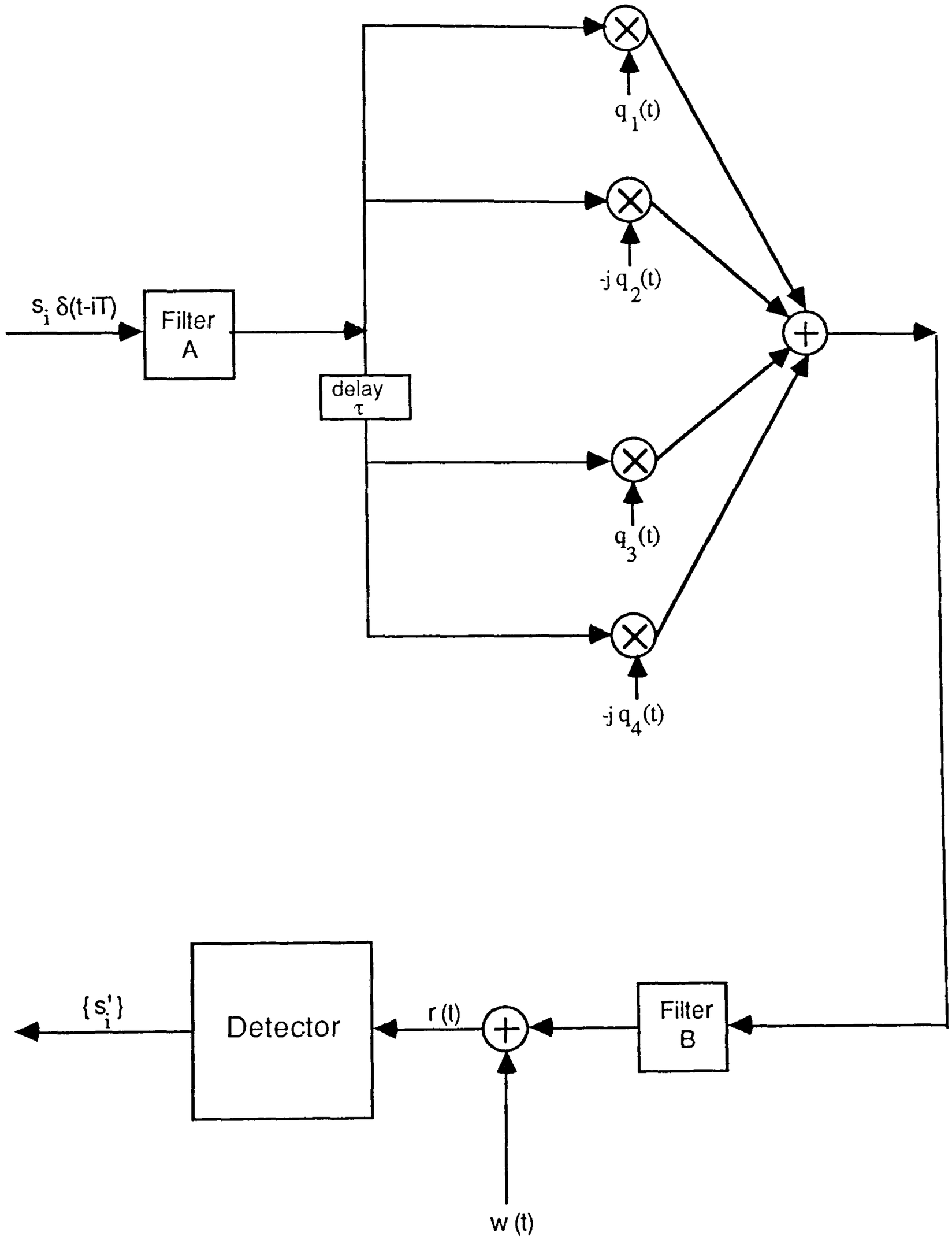


Fig. 3.3.5 Baseband model of data transmission system using QAM over HF radio links

CHAPTER 4

NEAR MAXIMUM LIKELIHOOD DETECTORS FOR A 4.8 KBIT/S HF RADIO MODEM

4.1 INTRODUCTION

For a sequence of data symbols transmitted over a non-ideal, bandlimited channel which introduces intersymbol interference and additive noise, the optimum detection process is one which minimises the probability of error in the detection of the entire transmitted sequence, given that the whole sequence has been received [1-3]. It is assumed here that the transmitted symbols have a finite alphabet.

The optimum detector is said to be that which performs a process of maximum likelihood estimation [1] of the transmitted sequence, and is known as a maximum likelihood detector, or, maximum likelihood sequence estimator [1,4].

One such realization of a maximum likelihood sequence estimator is that proposed by Forney [4], which uses a recursive algorithm called the Viterbi algorithm [5,6]. This is a relatively simple implementation of an optimum detection process which lends itself well to theoretical analysis and performance evaluation. It is optimum in the sense that it provides a maximum likelihood estimate of the entire transmitted sequence, subject to certain conditions which will be explained in due course. This detector, which will be referred to in the thesis as the Viterbi detector, does not, however, minimize the probability of error in the detection of an individual data symbol. Nevertheless, it has been proven that, at high signal-to-noise ratios, the performance of the Viterbi detector is similar to that of any other detector whose optimization criterion is the minimization of the probability of error in an individual data symbol [1,4].

The detection processes studied in this chapter are based on the Viterbi detector. As such, the chapter begins with a description of the Viterbi algorithm. The theory of maximum likelihood detection is however, not included, since it is widely available in the published literature [1-3], and it is the author's view that no loss of logicality results due to this omission. It should also be noted that the Viterbi algorithm is explained, in Section 4.2, from the viewpoint of its operation in a Viterbi detector, rather than in a more general sense. The constraints and limitations of the Viterbi detector are explained in Section 4.3, and a need for near maximum likelihood detectors is established. Section 4.4 gives a model of the data transmission system that was used in the simulation tests, and

this is followed in Section 4.5, by a detailed study of the near maximum likelihood detectors that were tested. In Section 4.6, the results are discussed, and the performance of the detectors are compared and contrasted. The major conclusions are summarized in Section 4.7.

4.2 THE VITERBI ALGORITHM

The Viterbi algorithm was originally developed in 1967, for the decoding of convolutional codes [5]. However, it was later observed to be a recursive optimal solution to the problem of estimating the state sequence in a finite state, discrete time, first order Markov process observed in memoryless noise [6]. The process should be Markov in the sense that the probability of being in a particular state at time j, given all the states up to that time, depends only on the state at time j-1. The process is said to be observed in memoryless noise in the sense that the observations at time j, depend probabilistically, only on the state transitions at that particular time instant. Thus, if a process could be represented by a finite state shift register model, (subject, of course, to the memoryless noise observation constraint), it is amenable for processing by the Viterbi algorithm.

The objective in this section is to explain the Viterbi algorithm in terms of its operation as a maximum likelihood detector. Toward this end, consider the model of the data transmission system given in Fig. 4.2.1. This is a model of a synchronous, serial system, where the element rate is 1/T bauds. The signal at the input to the transmitter filter is assumed to be a set of impulses, regularly spaced at intervals of T seconds, that is,

$$s(t) = \sum_i s_i \, \delta(t-iT) \qquad (4.2.1)$$

The $\{s_i\}$ are assumed to be statistically independent and equally likely to have one of m possible values. The transmitter filter has an impulse response a(t), and is the filter which shapes the spectrum of the input signal, in order that it matches the available bandwidth of the transmission path. The transmission path has an impulse response h(t), which is assumed to be of finite duration and time invariant. White Gaussian noise with zero mean and a two-sided power spectral density of $N_0/2$ is added at the output of the transmission path. The receiver filter has an impulse response b(t), and its function is to remove the noise frequencies which lie outside the signal bandwidth. For the purpose of this analysis, the absolute value of the receiver filter transfer function, |B(f)|, is assumed to have a rectangular shape, as shown in Fig. 4.2.2. The transmitter filter, transmission path and

receiver filter in cascade, is said to form a linear baseband channel with impulse response y(t), such that,

$$y(t) = a(t) * h(t) * b(t) \qquad (4.2.2)$$

The received signal, r(t), at the output of the receiver filter, is sampled once per signal element, at the time instants t = iT, (i = 0,1,.....), to give the received signal sequence $\{r_i\}$. The $\{r_i\}$ are fed to the detector, which gives at its output, the detected data symbol sequence $\{s_i'\}$.

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

$$r(t) = \sum_i s_i\, y(t-iT) + w(t) \qquad (4.2.3)$$

where,

$$w(t) = n(t) * b(t) \qquad (4.2.4)$$

The power spectral density of n(t) is $N_0/2$ and hence the power spectral density of w(t) is $N_0 |B(f)|^2 / 2$. Therefore, the autocorrelation function, $R_w(\tau)$, of w(t), is given by

$$R_w(\tau) = \int_{-\infty}^{\infty} \frac{1}{2} N_0 \, |B(f)|^2 \, e^{j 2\pi f \tau} \, df \qquad (4.2.5)$$

Substituting for |B(f)| (as shown in Fig.4.2.2),

$$R_w(\tau) = \frac{1}{2} N_0 \int_{-1/2T}^{1/2T} e^{j 2\pi f \tau} \, df$$

$$= \frac{N_0}{2T} \frac{\sin(\pi\tau/T)}{\pi\tau/T} \qquad (4.2.6)$$

The variance of w(t) is,

$$R_w(0) = \frac{N_0}{2T} \qquad (4.2.7)$$

w(t) is, thus, a bandlimited Gaussian noise waveform with zero mean and variance $N_0/2$.

The signal at the output of the sampler, at time t = jT, is,

$$r(jT) = \sum_i s_i\, y(jT-iT) + w(jT)$$

$$= \sum_i s_i\, y([j-i]\,T) + w(jT) \qquad (4.2.8)$$

Thus, the j^{th} received sample is,

$$r_j = \sum_{h=0}^{g} s_{j-h}\, y_h + w_j \qquad (4.2.9)$$

where $y_h = y(hT)$ and $Y_h = 0$ for $h < 0$ and $h > g$. Thus, the samples y_h, (h=0,1,....g), form a finite sequence and is represented by the (g+1)-component vector Y, where

$$Y = y_0 \quad y_1 \quad ..... \quad y_g \qquad (4.2.10)$$

Y is called the sampled impulse response of the linear baseband channel. It is of finite length and is assumed to be completely known at the receiver.

The noise sample w_j is a sample value of the noise function w(t) at time t=jT. Since w(t) was shown to be a bandlimited Gaussian noise waveform with zero mean and variance $N_o/2T$, so the $\{w_i\}$ are a set of Gaussian random variables with zero mean and variance $N_o/2T$. Moreover, it could be shown from eqn. 4.2.6, that the $\{w_i\}$, for different i, are uncorrelated [7,8], and since they have zero mean, are statistically independent [1] as well. This assumption of statistical independence of the noise samples entering the detector, is extremely vital for the operation of the Viterbi algorithm. This is the reason for the choice of the receiver filter as given in Fig. 4.2.2.

It could be seen from eqn. 4.2.9, that the model given in Fig. 4.2.1 could be

represented, equivalently, by a discrete time, feedforward transversal filter with tap gains y_0 y_1....y_g, to whose output is added a noise sample. This equivalent model is shown in Fig. 4.2.3. Since the $\{s_i\}$ are assumed to have only a finite number, m, of possible values, this is essentially a finite state, shift register model. Moreover, since the additive noise affects only the current sample, the process could be said to be observed in memoryless noise. Thus, the model fits in perfectly with the requirements observed at the beginning of this section, and the Viterbi algorithm could now be used for the estimation of its state sequence, i.e. for the estimation of the transmitted data symbol sequence.

Let,

$$S_k = s_1 \quad s_2 \quad ..... \quad s_k \tag{4.2.11}$$

$$W_k = w_1 \quad w_2 \quad ..... \quad w_k \tag{4.2.12}$$

and

$$R_k = r_1 \quad r_2 \quad ..... \quad r_k \tag{4.2.13}$$

S_k, W_k and R_k are k-component vectors whose i^{th} components are, respectively, s_i, w_i and r_i for i = 1, 2, ... k. They represent, respectively, the symbol values, noise components and received samples up to time kT, in the transmitted message. Clearly, S_k is equally likely to have any one of its m^k possible values. Now let,

$$X_k = x_1 \quad x_2 \quad ..... \quad x_k \tag{4.2.14}$$

$$Z_k = z_1 \quad z_2 \quad ..... \quad z_k \tag{4.2.15}$$

and

$$U_k = u_1 \quad u_2 \quad ..... \quad u_k \tag{4.2.16}$$

X_k, Z_k and U_k are k-component vectors whose i^{th} components are x_i, z_i and u_i. x_i is a possible value of s_i and as such, may have any one of m values. z_i is a possible estimate of the signal component in r_i, and is given by,

$$z_i = \sum_{h=0}^{g} x_{i-h}\, y_h \qquad (4.2.17)$$

u_i is a possible value of w_i, satisfying

$$r_i = z_i + u_i \qquad (4.2.18)$$

Clearly,

$$u_i = r_i - z_i \qquad (4.2.19)$$

and hence U_k is the unitary distance between R_k and Z_k, where,

$$|U_k|^2 = |u_1|^2 + |u_2|^2 + \ldots\ldots + |u_k|^2 \qquad (4.2.20)$$

Under the conditions assumed, (most importantly, the equally likely condition for S_i and the statistical independence condition for the Gaussian noise samples w_i), the maximum likelihood estimate of the transmitted sequence is given by the vector $(X_k)_{max}$, where this is the value of the vector X_k such that $|U_k|^2$ is minimized. $(X_k)_{max}$ is also the possible value of S_k that is most likely to be correct [1].

The Viterbi algorithm is now explained as follows. Just prior to the receipt of the sample r_{k+1}, the detector holds in store m^g vectors, $\{X_k\}$, corresponding to the m^g different possible combinations of their last g components, i.e., x_{k-g+1}, x_{k-g+2}, ... x_{k-1}, x_k. These set of vectors are called 'survivors' corresponding to the state given by that particular combination of its last g components. Each survivor is associated with a 'cost', $|U_k|^2$, as given by eqns. 4.2.17, 4.2.19 and 4.2.20, and is the vector with the lowest cost, corresponding to its own state. These costs are stored alongside the survivors. It should be noted here, that these costs are analogous to the 'metrics' which need to be computed in the traditional exposition of the Viterbi algorithm [6], the difference (apart from its value) being that, in the case of metrics, the algorithm searches for maximum values, whereas in this case, the algorithm attempts to find minimum costs.

When r_{k+1} is received, each of the m^g survivors is expanded m ways to give m^{g+1} vectors, $\{X_{k+1}\}$, where the last component, x_{k+1}, takes on its m different possible

values, and the first k components remain unchanged. The cost of these vector $\{X_{k+1}\}$ is given by $|U_{k+1}|^2$, where,

$$|U_{k+1}|^2 = |U_k|^2 + \left\{ r_{k+1} - \sum_{h=0}^{g} x_{k+1-h}\, y_h \right\}^2 \qquad (4.2.21)$$

The resulting $(m)^{g+1}$ costs are stored alongside the corresponding $(m)^{g+1}$ vectors, where the costs are evaluated using the stored values of $|U_k|^2$.

Now, for each of the m^g possible combinations of the last g components of X_{k+1}, (that is, x_{k-g+2}, x_{k-g+3}, ... x_{k-1}, x_k, x_{k+1}), the detector selects the vector which has the smallest value of $|U_{k+1}|^2$. When this process is completed, the detector is left with the resulting m^g vectors $\{X_{k+1}\}$, which are of course, the survivors at time (k+1)T. The associated cost, $|U_{k+1}|^2$, is stored alongside the corresponding vector X_{k+1}. The maximum likelihood vector, $(X_{k+1})_{max}$, is now chosen as the vector with the lowest cost amongst the survivors. The process continues this way, on receipt of the signal r_{k+2} .

Ideally, the detection is carried out on receipt of the whole message, whereby, the components $\{x_i\}$ in $(X_k)_{max}$ are taken to be the detected values of the data symbols $\{s_i\}$, assuming of course, the duration of the entire message to be kT seconds. In practice, (especially if the message is long and k is a large positive integer), this would involve an excessive amount of storage. Thus, an appropriate delay of n time intervals is introduced before the detection of any of the s_i, and instead of storing m^g k-component vectors $\{X_k\}$, the detector stores m^g n-component vectors $\{Q_k\}$, where,

$$Q_k = x_{k-n+1} \quad x_{k-n+2} \quad \ldots\ldots \quad x_k \qquad (4.2.22)$$

It could be seen that Q_k is formed by the last n components of the corresponding vector X_k. s_{k-n+1} is now detected as the value of x_{k-n+1} in the maximum likelihood vector $(Q_k)_{max}$, corresponding to $(X_k)_{max}$.

If n is chosen to be large enough, this truncation process rarely reduces the optimality of the detector, since, in almost every case, the segment, $x_1\ x_2\ \ldots\ x_{k-n}$, becomes common to all the survivors. Thus, the true maximum likelihood vector $(X_k)_{max}$, is always held in the detector store. Moreover, in the rare instances that this is not so, it

has been observed that the effect on the performance of the detector is negligible [6]. As a general rule however, n is always chosen to be greater than g+1 [1] .

The recursion may be started, in practice, by making one of the m^g stored vectors exactly equally to the first g data symbols, and assigning to it, a zero cost. The remaining $m^g - 1$ vectors may be chosen arbitrarily with a very high value of cost assigned to each of them. After g recursion cycles, all of the arbitrarily chosen vectors would have been discarded, and every one of the new set of m^g survivors would have originated from the vector to which was initially assigned a zero cost.

As regards computational complexity, the Viterbi detector requires, for every recursion cycle, m^{g+1} squaring operations, m^g comparisons and m^{g+1} additions. In terms of storage, it requires $2m^g$ storage locations, half of which are used to store the survivors and the other half, to store their corresponding costs.

The symbol error probability, P(e), for the Viterbi detector, has been shown to be lower and upper bounded as [4]

$$k_0 Q(\frac{d_{min}}{2\sigma}) \leq P(e) \leq k_1 Q(\frac{d_{min}}{2\sigma}) \qquad (4.2.23)$$

where d_{min} is the minimum euclidean distance between any two possible $\{Z_k\}$ as given in eqn. 4.2.15, σ^2 is the variance of the noise samples $\{w_i\}$, k_o and k_1 are two small constants and Q(.) is the complementary error function. It has also been shown that d_{min} is a function of both the energy of the sampled impulse response of the channel, and the number of components it possesses [9]. Thus, given these values, it should be possible to predict the performance of the Viterbi detector.

4.3 MODIFICATIONS AND DERIVATIVES OF THE VITERBI DETECTOR

It should be recalled, at this stage, that, for the Viterbi detector to perform true maximum likelihood detection, the noise components in the received signal samples at the input to the detector should be uncorrelated. (It was shown by Ungerboeck [10] that this need not necessarily be the case and that, if the noise entering the detector were correlated, maximum likelihood detection could still be performed. However, this required a fundamental modification to the Viterbi algorithm, and the detectors given in the thesis do

not follow this path.) In the model described in the previous section, this was acheived by choosing the receiver filter such that it had a flat amplitude characteristic with sharp cut-offs, as shown in Fig. 4.2.2. Alas, this type of filter could never be designed in practice since it would require an infinite number of taps. Thus, any practical filter used would correlate the noise, preventing true maximum likelihood detection.

When Forney originally proposed the optimum receiver, [4] the above mentioned problem was circumvented by replacing the receiver filter with a matched filter, sampling the output of the matched filter, and feeding these samples to a linear transversal filter, which is used just ahead of the detector (see Fig.4.3.1). When this linear transversal filter is designed according to certain criteria [4], the noise components of the signals at its output become uncorrelated, i.e., this filter essentially decorrelates the noise. Moreover, the outputs of this filter (which are also the inputs to the detector), have been shown to form a set of sufficient statistics for the maximum likelihood detection of the transmitted data signals. The cascade of the matched filter, sampler, and linear transversal filter is known as the 'whitened matched filter'.

The criteria under which the linear transversal filter becomes part of the whitened matched filter, will now be explained. It should be noted that no proof will be given for any results and observations, (the interested reader should refer to [4] for such), rather, the main points of concern will be summarized.

Consider Fig. 4.3.1. g(t) here, is the impulse response of the cascade of the transmitter filter and transmission path. The receiver filter is a filter matched to g(t), (i.e., it has an impulse response g(-t)), such that the signal-to-noise ratio at its output is maximised. The samples at the output of this receiver filter, $\{r_i'\}$, form a set of sufficient statistics for the estimation of the transmitted data sequence.

Now, let the z-transform of the sampled impulse response of the channel (which includes the transmitter filter) and receiver filter, by $Y'(z)$. It could be shown that $Y'(z)$ is a polynomial in z with $2l+1$ terms, has $2l$ complex roots, and these roots occur in complex conjugate reciprocal pairs [4]. Let these pairs be ρ_1 and $1/(\rho_1)^*$, ρ_2 and $1/(\rho_2)^*$, and so on, until ρ_l and $1/(\rho_l)^*$. It could then be seen that $Y'(z)$ may always be factored as

$$Y'(z) = Y_1(z)\, Y_2(z) \qquad (4.3.1)$$

where $Y_1(z)$ is a polynomial in z with the l roots $\rho_1, \rho_2, \ldots \rho_l$, and $Y_2(z)$ is a polynomial

in z with the l roots $1/(\rho_1)^*$, $1/(\rho_2)^*$, ... $1/(\rho_l)^*$. If now, the tap gain coefficients of the linear transveral filter are given by the coefficients of the polynomial

$$F(z) = Y_1^{-1}(z) \qquad (4.3.2)$$

then the noise components of the received samples at the output of this linear filter will be uncorrelated. Moreover, these received samples themselves would form a sufficient set of statistics for the estimation of the transmitted data sequence.

$Y_1(z)$ here, is assumed to be chosen such that all of its roots lie outside the unit circle. All the roots of $Y^{-1}(z)$ would then lie inside the unit circle, making the linear transversal filter a physically realisable, stable, discrete time filter. (Of course, the roots on the unit circle could theoretically be accommodated by removing the stability condition of the linear filter - details of this general case are given in ref. [4]). A little thought would now indicate that the resultant z-transform of the channel and the whitened matched filter, has no zeros outside the unit circle, these having been eliminated by the linear filter. Such are the characteristics and operations of the linear transversal filter which is used to decorrelate the noise.

In the explanation in Section 4.2, of the operation of the Viterbi detector, no mention was made of a whitened matched filter. Thus, it may appear on first thought, that the signals at the input to the detector (see Fig. 4.2.1) do not form a set of sufficient statistics for the estimation of the transmitted data sequence, and that the noise components in these signals are correlated, thereby preventing maximum likelihood detection. However, it has been shown that, when the channel is strictly bandlimited to -1/2T to 1/2T Hz., (i.e., when the channel is used at the Nyquist rate), the whitened matched filter degenerates into a lowpass filter with a flat amplitude response, arbitrary phase characteristic, and a cutoff frequency at half the symbol rate, 1/2T, where the cutoff is assumed to be instantaneous [11]. It is this filter that is assumed in Fig. 4.2.1 and shown in Fig. 4.2.2, and hence, the validity of the Viterbi detector as a maximum likelihood detector still holds, as does the subsequent analysis in Section 4.2. The noise components at the output of this lowpass filter are statistically independent, and the outputs themselves are said to form a set of sufficient statistics for the estimation of the transmitted data sequence.

An added convenience here, is to make the resultant sampled impulse response of the transmitter filter, transmission path and receiver filter, minimum phase [1], whence the effective number of components of this sampled impulse response would be a

minimum, thereby reducing the complexity of the Viterbi detector. It has been shown that this can be accomplished by specifying the phase characteristic of the receiver filter to be linear, and following this receiver filter with a sampler and a linear feedforward transversal filter (that which forms the first part of a conventional nonlinear equalizer [1,12]). The linear feedforward transveral filter here, is an allpass network which replaces the roots of the z-transform of the channel which lie outside the unit circle, by the complex conjugates of their reciprocals, so that the resultant z-transform of the channel, receiver (lowpass) filter, sampler and linear feedforward transversal filter, has no roots outside the unit circle. It should also be noted that the linear feedforward transversal filter does not change either the noise statistics, or any amplitude distortion that may have been introduced by the channel, performing instead, a pure phase transformation of the signal at its input.

Although an implementable design, the Viterbi detector does have a rather restrictive, if not prohibitive, characteristic, in that the computational complexity grows exponentially with the number of components in the sampled impulse response of the channel. Indeed, when the sampled impulse response of the channel has more than a few components, the complexity becomes unmanageable. One solution to this problem is to shorten the channel impulse response by introducing some form of pre-filtering of the signal at the input to the detector, such that the detector sees only a 'desired channel impulse response', which could be constrained to be of very short length [13-16]. However, this prefiltering tends to correlate the noise, making the detection process suboptimal. Another approach is to replace this prefilter with a nonlinear equalizer which partially equalizes the channel before passing its output onto the detector [17]. Again, this technique suffers from the error extension effects common to the nonlinear equalizer and also reduces the effective signal-to-noise ratio. Thus, both these solutions result in detectors which are very much sub-optimal.

An alternative method of reducing the complexity of the Viterbi detector when a large number of components are present in the channel sampled impulse response, is to reduce the number of vectors held in the detector store at any particular time instant. Thus, instead of holding m^g survivors, the detector may hold m' survivors, ($m' < m^g$), where these are chosen according to some criterion [18,19]. These class of detectors are known as reduced state Viterbi detectors, and their performance approaches that of an optimum detector as the noise variance approaches zero. At low and moderate signal-to-noise ratios however, these detectors are suboptimum, with the performance degradation relative to the optimum detector being dependent on the ratio of m' / m^g, the lower the ratio, the higher the degradation.

Near maximum likelihood detectors are a class of reduced-state Viterbi detectors

where m' is generally much smaller than m^g and where all the m' stored vectors need not be survivors. The criterion for the choice of the m' vectors is such that the degradation in tolerance to noise is kept to a minimum. Several such criteria have been established in the published literature [12,20-27]. The detectors employed in the ensuing work are all near maximum likelihood detectors and are known in the sequel as System A, System B and System C. In order to explain the operation of these detectors, it is first necessary to describe the model of the data transmission system and its attendant parameters that were used in the simulation.

4.4 MODEL OF THE DATA TRANSMISSION SYSTEM USED IN THE SIMULATION

The model of the data transmission system is given in Fig. 4.4.1. This is based on the baseband model of the QAM system as given in Fig. 3.3.5, with a few necessary changes, as will be explained below. These changes are incorporated in order to model, as closely as possible, a practical data modem within the framework of the stated objective, i.e., that of assessing the performance of various near maximum likelihood detectors when operating over HF radio links.

The model itself assumes a synchronous, serial data transmission system employing a 4-level QAM signal, with an element rate of 2400 bauds, resulting, of course, in an information rate of 4800 bit/s. The information to be transmitted is in the form of a sequence of binary digits $\{\alpha_i\}$, where these are statistically independent and equally likely to have either of the values

$$\alpha_i = 0 \text{ or } 1 \qquad (4.4.1)$$

These $\{\alpha_i\}$ are fed to a differential encoder, which gives at its output, the differentially encoded signal that will be transmitted over the HF radio link. Differential encoding of the input sequence is carried out for the following reason. When the transmitted signal is in a deep fade, the carrier experiences large and rapid phase changes called phase jumps. These phase jumps are manifested in the equivalent, complex valued, linear baseband channel model, as rotations of multiples of $\pi/4$ radians (in the complex number plane), in each of the complex valued received signal samples. This, of course, would result in a string of errors in the detected data symbol values, where the occurrence of these errors would continue until the end of the transmitted message, even in the absence of noise. However, these prolonged error bursts can be prevented by coding the difference in phase

between two consecutive symbol values (as represented in the complex number plane), where the transmitted signal would now represent a phase rotation. By employing the corresponding differential decoding process at the receiver, the original symbols, $\{\alpha_i\}$, could be reconstructed. The mechanics of the differential encoding and decoding processes are given in Appendix B. It should be noted however, that in this particular model, perfect estimation of the channel is assumed, and as such, the phase jumps that occur are known at the receiver, making differential coding seem rather superfluous. Nevertheless, in any practical modem operating over a fading channel, differential coding is an essential constituent part, (since perfect estimation can no longer be assumed); hence its inclusion here.

The output of the differential encoder (see Appendix B) is a sequence of data symbol values, $\{s_i\}$, in the form of impulses occurring at 2400 symbols/s, where

$$s_i = s_{1,i} + js_{2,i} \qquad (4.4.2)$$

with $s_{1,i} = \pm 1$ and $s_{2,i} = \pm 1$. Each s_i is thus one of a set of 4-level data symbols as shown in Fig. 3.2.2, and it is assumed that $s_i = 0$ for $i \leq 0$, so that s_i is the i^{th} transmitted data symbol value. Moreover, the $\{s_i\}$ are statistically independent and equally likely to have any one of their four possible values.

The impulses, $\{s_i\,\delta(t\text{-}iT)\}$, where $T = 1/2400$, are fed to a channel model comprising the transmitter filter, HF radio link, and receiver filter. The model here, is a scaled, minimum phased version of the equivalent, time varying, linear baseband channel that was derived in Section 3.3 (shown in Fig. 3.3.5) and will be explained in more detail shortly. The complex valued, time varying, impulse response of the channel is $y_i(t)$, and the stream of signal elements at the output of the channel are given by $\{s_i\,y_i(t\text{-}iT)\}$. Stationary, complex valued, zero mean, coloured, Gaussian noise w(t), is added to the signal at the output of the channel, to give the complex valued received waveform r(t). Again, the reasons for, and consequences of, the noise being correlated, will be given in the detailed explanation of the channel model.

The received waveform r(t), is now sampled once per data symbol, at the time instants {iT}. The delay in transmission is neglected, such that the first nonzero sample of the received signal waveform is assumed to arrive with no delay. The sample r_i, of the received waveform at $t = iT$, is now given by,

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_{i,h} + w_i \qquad (4.4.3)$$

where,

$$y_{i,h} = y_{i-h}(hT) \qquad (4.4.4)$$

and

$$w_i = w(iT) \qquad (4.4.5)$$

It is assumed that $y_{i,h} = 0$ for $h < 0$ and $h > g$, and the vector

$$Y_i = y_{i,0} \quad y_{i,1} \quad \ldots\ldots \quad y_{i,g} \qquad (4.4.6)$$

is taken to be the sampled impulse response of the channel at t=iT.

The received samples, $\{r_i\}$, are next fed to the detector, which gives at its output, the detected symbol values $\{s'_{i-n}\}$. In the particular case considered here, the detector is assumed to have an exact knowledge of the channel (i.e., perfect estimation is assumed). The detector also introduces a delay of nT seconds such that the detected data symbol obtained on receipt of r_i, is s'_{i-n}. These detected data symbols, $\{s'_{i-n}\}$, are finally fed to a differential decoder, which performs an inverse operation to that of the differential encoder, and gives at its output, the detected binary digit sequence $\{\alpha'_{i-n}\}$.

It is now appropriate to discuss in further detail, the scaled minimum phased version of the equivalent linear baseband channel model. Toward this end, consider Fig. 4.4.2. This represents the time varying linear baseband channel whose impulse response is $y_i''(t-iT)$, and is exactly that derived in Section 3.3 and given by eqn. 3.3.33. It should be noted however, that the model of the HF radio link itself could comprise either two or three skywaves, with parameters corresponding to those of Channel 1, Channel 2 and Channel 3, as given in Table 2.5.2. In the sequel, the two skywave model is assumed.

From eqns. 3.3.8, 3.3.25 and 3.3.33,

$$y_i''(t\text{-}iT) = \{ a(t\text{-}iT)\,[q_1(t) - jq_2(t)] + a(t\text{-}\tau\text{-}iT)\,[q_3(t) - jq_4(t)] \} * b(t) \quad (4.4.7)$$

where,

$$a(t) = a'(t) * g(t)\, e^{-j2\pi f_c t} \quad (4.4.8)$$

and

$$b(t) = [\, d(t) * c(t)\,]\, e^{-j2\pi f_c t} * b'(t) \quad (4.4.9)$$

τ here, is the relative delay between the two skywaves and f_c is the carrier frequency of the QAM signal, where

$$f_c = 1800 \text{ Hz.} \quad (4.4.10)$$

$a'(t)$, $c(t)$ and $b'(t)$ are the impulse responses of the modem filters and $g(t)$ and $d(t)$ are the impulse responses of the radio filters. The attenuation and phase characteristics of these filters in the passband of the QAM signal, along with the sampled versions of $a(t)$, $a(t\text{-}\tau)$ and $b(t)$, (for τ=1.1ms., 2ms. and 3ms.), are given in Figs. 4.4.3 - 4.4.5 and Tables 4.4.1 - 4.4.2. The sampled impulse responses assume a sampling rate of 4800 samples/s, the reason for which will become clear shortly.

The values given in Tables 4.4.1 - 4.4.2 have been obtained from Dr. S.N. Abdullah, and further details regarding the design of these filters are available in refs. [7] and [29]. However, it should be noted here, that each of the sampled versions of $a(t)$, $a(t\text{-}\tau)$ and $b(t)$, (for τ = 1.1ms., 2ms. and 3ms.), are minimum phase (or near minimum phase), so that the cascade of any one transmitter filter and the receiver filter would result in a minimum phase (or near minimum phase) sequence. Thus, in the absence of any fading, the channel formed by the transmitter and receiver filters is minimum phase (or near minimum phase).

The sampled impulse response of the channel (as given in eqn. 4.4.6), is obtained by sampling $y_i''(t\text{-}iT)$ in eqn. 4.4.7 at the rate of 2400 samples/s (i.e., 1/T), which is the baud rate of the system. However, the convolution itself (in eqn. 4.4.7) is carried out at 4800 samples/s, and as such, the sampled impulse response is obtained by appropriately choosing every other sample in the resultant sampled, convolved sequence.

The higher sampling rate of 4800 samples/s is well above the Nyquist rate for the filters a(t) and b(t) and as such, ensures that no information is lost in the discretization of these filters.

Now, let $\{a_{1,k}\}$, $\{a_{2,k}\}$ and $\{b_k\}$ be the sequences obtained by sampling a(t), a(t-τ) and b(t) at 4800 samples/s. $\{a_{1,k}\}$ and $\{b_k\}$ here, correspond to the two sequences given in Table 4.4.1, while $\{a_{2,k}\}$ could be any one of the sequences in Table 4.4.2. Also, let $\{q_{1,2i}\}$, $\{q_{2,2i}\}$, $\{q_{3,2i}\}$ and $\{q_{4,2i}\}$ be the sequences obtained by sampling the four fading components $q_1(t)$, $q_2(t)$, $q_3(t)$ and $q_4(t)$, at 4800 samples/s, where the samples in each sequence correspond to those obtained upto the time instant t=iT. Then, from eqns. 4.4.7 and 4.4.4, it could be seen that the components of the sampled version of $y_i''(t-iT)$, at t=iT, are given by,

$$y''_{i,h} = \frac{T}{2}\left\{\sum_{k=0}^{2h}\left[a_{1,k}\left[q_{1,2(i-h)+k} - jq_{2,2(i-h)+k}\right] + a_{2,k}\left[q_{3,2(i-h)+k} - jq_{4,2(i-h)+k}\right]\right] b_{2h-k}\right\} \qquad (4.4.11)$$

for h = 0,1, ... g, where g is dependent on the maximum delay between the first and the last (which, for a two skywave model, is the second) skywave. The values of (g+1) for Channels 1, 2 and 3 are, respectively, 22, 22 and 20.

In eqn. 4.4.11, the $\{y''_{i,h}\}$ are automatically obtained at 2400 samples/s since the equation incorporates the fact that every other sample is chosen. It also takes into account the relationship in eqn. 4.4.4, thus ensuring the proper correspondence between the fading samples and the filter samples. The multiplication by the inverse of the sampling rate of the $\{a_k\}$ and $\{b_k\}$ is necesssary in order to ensure that eqn. 4.4.11 is an exact representation, in the discrete time domain, of eqn. 4.4.7.

It is now attempted to make the sequence $\{y''_{i,h}\}$, (for h = 0, 1, ... g), a minimum phase sequence, by passing it through a minimum phasing network. A minimum phased channel would have most of its energy concentrated in the first few components. Although not critical in a full Viterbi detector, minimum phasing of the channel is desirable when employing near maximum likelihood detectors, (essential in the case of a conventional nonlinear equalizer [1]), in order that the cost functions give a reliable

indication (especially in fairly low noise situations), as to which vectors should be held in store.

In practice, minimum phasing would be carried out by employing an adaptive linear filter just ahead of the detector, where the function of this filter is to replace the zeros of the z-transform of the sampled impulse response of the channel which lie outside the unit circle, (i.e., those that have an absolute value greater than unity), by the reciprocals of their complex conjugates [1,28]. However, such a filter, in its ideal form, would require an infinite number of taps, and as such, in order to preserve the practical feasibility of the model, it was decided that the minimum phasing network should operate only on those roots that have absolute values greater that d, where d>1. By increasing the value of d, a reduction in the number of taps required in a practical adaptive linear filter could be obtained. However, this cannot be carried on ad infinitum, since an increase in the value of d would also be accompanied by a corresponding degradation in the performance of any near maximum likelihood detector, due now, to the increasing departure of the channel from being minimum phase. Hence, the smallest practically feasible value of d should be employed, and this has been shown to be 1.05 [27].

Thus, the minimum phasing network in the assumed model operates on the roots of the z-transform of the sequence $\{y''_{i,h}\}$, (h = 0,1, ... g), that have absolute values greater than 1.05, by replacing these roots by the reciprocals of their complex conjugates. The network here comprises a root finding software module from the Numerical Algorithm Group library, (NAG - CO2ADF), which operates on the samples of $y_i''(t\text{-}iT)$ as given in eqn. 4.4.11, and some attendant calculations to determine the new 'channel' from the new set of roots. The impulse response at the output of the minimum phasing network is given by $y_i'(t\text{-}iT)$, where this is almost (though not strictly) a minimum phased channel. Thus,

$$y_i'(t\text{-}iT) \approx \text{min. phased } [y_i''(t\text{-}iT)] \qquad (4.4.12)$$

and,

$$\{ y'_{i,h} \} \approx \text{min. phased } [\{ y''_{i,h} \}] \qquad (4.4.13)$$

for h = 0, 1, ... , g .

The minimum phased sequence, $\{y'_{i,h}\}$, is now scaled such that its first

component is unity. This scaling is carried out purely for convenience in simulating the operation of the detectors. A similar scaling is performed on the noise sequence as well, in order not to introduce any undue change in the signal-to-noise ratio of the system. The impulse response of the scaled, minimum phased channel is given by $y_i(t\text{-}iT)$, and its sample values are given by $y_{i,h}$, where,

$$y_i(t\text{-}iT) \quad = \quad \text{scaled}\left[\, y_i'(t\text{-}iT) \,\right] \tag{4.4.14}$$

and

$$\{y_{i,h}\} \quad = \quad \left\{ \frac{y_{i,h}}{y_{i,0}} \right\} \tag{4.4.15}$$

for $h = 0, 1, \ldots g$. The sequence $\{y_{i,h}\}$ corresponds exactly to the vector Y_i in eqn. 4.4.6, and is the resultant 'sampled impulse response' vector of the time varying linear baseband channel. The vector Y_i is a minimum phase (almost, at least) sequence, whose first component is always unity.

The noise function w(t), which is added at the output of the channel model, (see Fig. 4.4.1), is represented in the discrete time domain by the sequence $\{w_i\}$, where w_i is the additive noise component in the sampled, received signal r_i, as given in eqn. 4.4.3. The $\{w_i\}$ are generated as follows. A sequence of statistically independent, complex valued, Gaussian random variables with zero mean and a variance of σ^2 in each of the real and imaginary parts, is generated via a software module (GO5DDF) from the NAG library. This sequence is next convolved with $\{b''_k\}$, where $\{b''_k\}$ is the sequence obtained by scaling the sequence $\{b_k\}$ such that $\sum |\, b''_k\,|^2 = 1$. Thus, these Gaussian random variables are passed through a scaled version of the same receiver filter that the signal passes through. (The $\{a_k\}$ and $\{b_k\}$ have been normalised such that the sum of the squares of each sequence is 2/T. All these normalization procedures are carried out in order to simplify the signal-to-noise ratio calculations, full details of which are given in Appendix A.3). It should be noted however, that the noise sequence obtained at the output of the scaled version of the receiver filter, would correspond to a sampling rate of 4800 samples/s, due to the $\{b_k\}$, (and hence the $\{b''_k\}$), being at this rate. Thus, it is necessary to choose every other sample at the output of the scaled receiver filter, thereby ensuring that the noise samples correspond to a sampling rate of 2400 samples/s, which is the baud rate of the system.

Let the noise sequence thus obtained, be represented by $\{w_i'\}$. This sequence is now scaled by dividing each component by $y'_{i,o}$, (in order to compensate for the similar scaling which was carried out on the minimum phased channel sequence), to give the sequence $\{w_i\}$, where,

$$\{w_i\} = \frac{\{w_i'\}}{\{y'_{i,0}\}} \qquad (4.4.16)$$

and is the discrete time representation of the noise function w(t).

It can be observed now that the noise samples $\{w_i\}$, are no longer uncorrelated, the correlation having been introduced by the scaled version of the receiver filter. This means that the noise components in the received signal entering the detector are correlated and as such, would prevent true maximum likelihood detection of the transmitted signal sequence. It is of course, possible now, to attempt to decorrelate the noise, but only at the expense of added complexity at the receiver ; a most unenviable option. Moreover, the objective is not to perform true maximum likelihood detection, (the usage of near maximum likelihood detectors eliminate such hopes anyway), but to develop and test near maximum likelihood detectors that are a good compromise between performance and complexity. As such, it was decided to maintain the status quo by leaving the correlated noise as it is, and accepting a slight degradation in the performance of the detectors. In the light of the results obtained, this matter is discussed further in Section 4.6.

Recalling the fact that the HF channel model on average does not introduce any gain or attenuation to the transmitted signal, (eqn. 3.3.44 and Section 2.6.2), the signal-to-noise ratio of the system is defined as

$$\psi = 10 \log_{10} \left(\frac{E_b}{\frac{1}{2} N_0} \right) \qquad (4.4.17)$$

where E_b is the average transmitted energy per bit at the output of the transmitter filter A in Fig. 4.4.2, and $N_o/2$ is the two-sided power spectral density of the real valued noise waveform n(t), in Fig. 3.3.1. Under the assumed conditions, that is, a 4-level QAM signal transmitted over the previously described channel model, ψ is shown in Appendix C (eqn. C.20) to be given by

$$\psi \quad = \quad 10 \log_{10} \left(\frac{\overline{s_i^2}}{2\sigma^2} \right) \tag{4.4.18}$$

where the numerator on the right hand side of the equation is the expected value of $| s_i |^2$ and σ^2 is the variance of the real or imaginary part of the discrete time noise sequence $\{w_i\}$.

4.5 THE DETECTORS

Three different near maximum likelihood detectors have been tested, namely, System A, System B and System C. These detectors operate on the data transmission system model shown in Fig. 4.4.1, i.e., that which was explained in Section 4.4.

4.5.1 System A

Just prior to the receipt of the sample r_i , the detector holds in store p different n-component vectors (sequences) $\{Q_{i-1}\}$, where,

$$Q_{i-1} \quad = \quad [\, x_{i-n} \quad x_{i-n+1} \quad ..... \quad x_{i-1} \,] \tag{4.5.1}$$

and x_{i-h} , (h = 1, 2, ... n), can take on any one of the four possible values of s_i , as given in eqn .4.4.2. It is assumed that $n \geq g$, where g is the number of components in the sampled impulse response vector given in eqn. 4.4.6. Each vector Q_{i-1} , is formed by the last n components of the (i-1)-component vector X_{i-1}, where,

$$X_{i-1} \quad = \quad [\, x_1 \quad x_2 \quad ..... \quad x_{i-1} \,] \tag{4.5.2}$$

Each vector X_{i-1}, represents a possible transmitted sequence of data symbols $\{s_j\}$, (j = 1, 2, ... i-1), and has associated with it, the cost

$$| U_{i-1} |^2 \quad = \quad \sum_{j=1}^{i-1} \left| r_j - \sum_{h=0}^{g} x_{j-h} y_{j,h} \right|^2 \tag{4.5.3}$$

where $x_i = 0$ for $i \leq 0$ and |u| is the absolute value of the scaler quantity u. $|U_{i-1}|^2$ is also taken to be the cost of the corresponding vector Q_{i-1} (which is obtained from the vector X_{i-1}). To simplify the terminology, these costs are denoted by,

$$c_{i-1} = |U_{i-1}|^2 \tag{4.5.4}$$

Under these conditions, and given the received sequence $\{r_j\}$, it can be shown that the vector X_{i-1} most likely to be correct, is that which has the smallest cost, (c_{i-1}), over all combinations of possible values of the $\{x_j\}$.

On receipt of the sample r_i, each of the p vectors $\{Q_{i-1}\}$, is expanded into four (n+1)-component vectors $\{P_i\}$, where,

$$P_i = [\, x_{i-n} \quad x_{i-n+1} \quad ..... \quad x_{i-1} \quad x_i \,] \tag{4.5.5}$$

Thus, there are four vectors $\{P_i\}$, originating from any one $\{Q_{i-1}\}$. The first n components of such a group of four, are as in the original vector Q_{i-1}, from which they were derived, and the last component x_i, takes on its four possible values given by the four possible values of s_i. Stored alongside each of the resulting vectors $\{P_i\}$, are their costs c_i, where,

$$c_i = c_{i-1} + \left| r_i - \sum_{h=0}^{g} x_{i-h} y_{i,h} \right|^2 \tag{4.5.6}$$

Using the fact that $y_{i,0} = 1$ (Section 4.4), eqn 4.5.6 could be rewritten as

$$c_i = c_{i-1} + \left| r_i - \sum_{h=1}^{g} x_{i-h} y_{i,h} - x_i y_{i,0} \right|^2$$

$$= c_{i-1} + \left| r_i - \sum_{h=1}^{g} x_{i-h} y_{i,h} - x_i \right|^2 \tag{4.5.7}$$

where c_{i-1} is the cost of the vector Q_{i-1}, from which P_i is derived.

The vector with the smallest cost is now chosen from the resulting set of 4p expanded vectors $\{P_i\}$. Let this chosen vector be denoted by P_s. The first component x_{i-n}, of P_s, is then taken as the detected value s'_{i-n}, of the data symbol s_{i-n}. Any vector P_i, whose first component differs from s'_{i-n}, is then discarded by assigning to it, an arbitrarily high value of cost. From the remaining vectors $\{P_i\}$ (which includes P_s), are selected the p vectors which have the smallest costs. The first component x_{i-n}, of all these selected vectors, is now omitted, to give the corresponding n-component vectors $\{Q_i\}$, where,

$$Q_i = [\, x_{i-n+1} \quad x_{i-n+2} \quad \ldots\ldots \quad x_{i-1} \quad x_i \,] \tag{4.5.8}$$

These p vectors $\{Q_i\}$ are now stored along with their associated costs, where these costs are the same as those of the $\{P_i\}$ from which the $\{Q_i\}$ were derived. The smallest of these costs is now subtracted from each of the p costs, so that the smallest cost becomes zero. This is done in order to avoid an unacceptable increase in the value of the costs $\{c_i\}$, over a long message, and it does not change the differences between the various costs. With the p stored vectors $\{Q_i\}$, and their costs $\{c_i\}$, the detector is ready for the next detection process, that is, the detection of s_{i-n+1}, on receipt of r_{i+1}.

The detection process just described involves the computation, per received data signal, of 4p costs and p searches, (i.e., p comparisons), through these 4p costs. In terms of memory, it requires 2p permanent storage locations, half of which are used to store the vectors, and the other half to store their corresponding costs. The delay in detection in System A is n sampling intervals. However, when n >> g, the effective delay is l sampling intervals, where l is the smallest positive integer for which x_{i-l} has the same value in all the p different stored vectors $\{Q_i\}$.

4.5.2. System B

System B has two versions, namely, the pseudobinary and the pseudoquaternary. The former involves two expansions per stored vector, and the latter involves four. (Indeed, System A has a pseudobinary version as well [12,23-26]; however, this is not one of the detectors that were tested). Since the pseudoquaternary version is a direct extension of the pseudobinary version, only the latter will be explained in detail. However, the main differences of the pseudoquaternary version will be noted at the end of this section.

The pseudobinary version of System B is a detector that operates as follows. Just prior to the receipt of the sample r_i , the detector holds in store p different vectors $\{Q_{i-1}\}$, together with their costs $\{c_{i-1}\}$, as in System A (eqns. 4.5.1 and 4.5.4). However, p must now be an even number and in addition to this, the vectors are arranged in pairs, where the two vectors in any one pair differ only in their last component x_{i-1} . This last component is such that the two vectors in any one pair of $\{Q_{i-1}\}$, are those with the smallest and second smallest cost, for the particular combination of values x_{i-n} , x_{i-n+1}, $\cdots$ x_{i-2} , in the given pair. Each pair of vectors is arranged such that the smallest cost vector is given by the first of the pair and the second smallest cost vector by the second of the pair. Thus, the p stored vectors $\{Q_{i-1}\}$, could be thought of as having originated from p/2 'root' vectors.

On receipt of the sample r_i , each vector Q_{i-1} is expanded into the corresponding vector P_i with the smallest cost, where P_i is as given in eqn. 4.4.5. For each of the vectors $\{Q_{i-1}\}$, the value of x_i which gives the corresponding smallest cost vector $\{P_i\}$, is determined by simple threshold level comparisons, [23,25] and does not involve the calculation of any costs. The detector next evaluates the costs of the p expanded vectors $\{P_i\}$, and selects from these the vector with the smallest cost. Let this vector be denoted as P_s . The detector then takes the value of x_{i-n} in P_s , as the detected value s'_{i-n} of the data symbol s_{i-n} . All vectors whose first component differs from s'_{i-n} are now discarded by assigning to them, an arbitrarily high value of cost. The first component of the remaining vectors, $\{P_i\}$, is omitted (without changing their costs), to give the corresponding n-component vectors $\{Q_i\}$.

From these remaining vectors $\{Q_i\}$, the detector now selects, (in addition to the smallest cost vector which originated from P_s), the $\{(p/2)-1\}$ vectors with the next smallest costs. This results in the detector having, at this stage, a total of p/2 vectors $\{Q_i\}$. To each of these vectors is then added an additional vector Q_i , whose first (n-1) components are the same as in the original vector Q_i , and whose last component, x_i , is such that the additional vector is that with the second smallest cost, for the given combination of values x_{i-n+1}, x_{i-n+2}, ... x_{i-1} , in the original vector Q_i . Again, simple algorithms have been developed to obtain the value of x_i , which gives the vector with the second smallest cost, [23,26], and as such, no costs need be evaluated at this stage.

The detector now holds in store p vectors, $\{Q_i\}$, which could be thought of as having originated from p/2 'root' vectors with the same values of x_{i-n+1}, x_{i-n+2}, ... x_{i-1}. Thus, the p vectors $\{Q_i\}$, could be held in pairs, as were the p vectors $\{Q_{i-1}\}$. The costs, $\{c_i\}$, of the p/2 vectors not yet determined, are next evaluated, and the smallest cost is subtracted from all the costs, in order to prevent any cost overflow. The p vectors $\{Q_i\}$, which are held in store along with their costs $\{c_i\}$, are now ready for the next detection process, that is, the detection of s_{i-n+1}, on receipt of r_{i+1}.

The pseudobinary version of System B involves the computation, per received data signal, of 1.5p costs, and 0.5p searches through p costs. The memory requirement, in terms of permanent storage locations, is the same as in System A, that is, 2p, and the delay in detection is n sampling intervals.

– The pseudoquaternary version of System B operates in a manner similar to the pseudobinary version, except that p now, is a multiple of four, and the initial stored vectors $\{Q_{i-1}\}$, are arranged in groups of four. Each group of four vectors are those which have the four smallest costs, for the particular combination of x_{i-n}, x_{i-n+1}, ···· x_{i-2}, in the given group. Thus, the p vectors $\{Q_{i-1}\}$ could be considered to have originated from p/4 'root' vectors. In the case considered here, that is, a 4-level QAM signal, the vectors with the four smallest costs are, of course, given by the four possible expansions of a 'root' vector.

On receipt of the signal r_i, the detection process proceeds exactly as in the pseudobinary version of System B, until the stage where the detector holds p vectors $\{Q_i\}$, which have originated from the p vectors $\{P_i\}$. The detector now selects, in addition to the smallest cost vector which originated from P_s, $\{(p/4) - 1\}$ vectors with the next smallest costs, to give a total of p/4 selected vectors $\{Q_i\}$. To each of these is added three more vectors, that is, those that give the second, third and fourth smallest costs, for the particular combination of x_{i-n+1}, x_{i-n+2}, ... x_{i-1}, in the corresponding original Q_i. Again, this is done by threshold level comparisons, and do not require the computation of any costs [26]. As before, in this particular case where a 4-level signal is used, the three additional vectors are given by the original vector Q_i, with its last component x_i, replaced by the three remaining possible values it could take.

The detector now has in store p vectors, $\{Q_i\}$, which could be thought of as having originated from p/4 'root' vectors. These vectors are arranged in groups of four,

as were the vectors $\{Q_{i-1}\}$ at the start of the detection process. The 0.75p costs, $\{c_i\}$, that have yet to be determined, are next evaluated. The smallest cost is subtracted from all the costs and the detector stores the p vectors $\{Q_i\}$, along with their costs $\{c_i\}$, ready for the next detection process.

The pseudoquaternary version of System B involves the computation, per received data signal, of 1.75p costs and 0.25p searches through p costs. The memory requirement (in terms of permanent memory locations), and the delay in detection, is the same as that in the pseudobinary version of System B.

4.5.3 System C

System C is a modified version of System B, where now, the number of expansions, η, of a given vector Q_{i-1}, is a variable. η is varied such that it reduces from 4 to 2 to 1 as the cost of a vector, Q_{i-1}, increases. This is best explained via the following example.

Let the detector initially hold in store p vectors, $\{Q_{i-1}\}$, where p=16. Each of these vectors is first expanded into the corresponding vector P_i, with the smallest cost. The detection of s_{i-n} is then carried out exactly as in System B, and the corresponding 16 vectors, $\{Q_i\}$, are obtained. From these 16 vectors $\{Q_i\}$, the detector selects the two vectors with the smallest and second smallest cost, and adds to each of them, three more vectors, as in the pseudoquaternary version of System B. Then, the detector selects, from the original set of 16 vectors $\{Q_i\}$, the two vectors with the third and fourth smallest cost, and adds to each of them, one more vector, as in the pseudobinary version of System B. Finally, the detector selects, from the original set of 16 vectors $\{Q_i\}$, the four vectors with the fifth to eighth smallest cost, but does not add any more vectors to these four. The eight vectors remaining from the original set of 16 vectors $\{Q_i\}$, are now discarded, and the detector is left with 16 selected vectors, (and of course, their costs), ready for the next detection process. If, following the detection of s_{i-n}, any vectors $\{P_i\}$, are discarded, this merely reduces the number of vectors $\{Q_i\}$, that are subsequently discarded at the end of each detection process, and does not, in any other way, change the detection process.

The complexity of System C depends not only on the number of stored vectors p, but also on the way in which the expansions are carried out. As such, quantification of this parameter is left until the next section, where a comparison of the various detectors

that were used, is given . Memory requirements (permanent storage), and the delay in detection, however, are the same as in the previous systems, namely, 2p storage locations and a delay in detection of n sampling intervals.

4.6 COMPUTER SIMULATION TESTS AND RESULTS

Computer simulation tests have been carried out to assess the performance of the various near maximum likelihood detectors mentioned in the previous section. The performance of a nonlinear equalizer has been included for comparisons sake. The model of the data transmission system used is that developed in Section 4.4, where a QPSK signal is employed to transmit the information at a rate of 4800 bit/s. The detectors have been tested over Channels 1, 2 and 3, whose parameters are given in Table 2.5.2. Before discussing the results however, it is first necessary to collate the assumptions made in the tests, in order for the results to be viewed in their proper perspective.

As mentioned in Section 4.4, it is assumed that the detectors have a perfect knowledge of the channel at every sampling instant, i.e., perfect estimation of the sampled impulse response of the channel is assumed. This is a major assumption in testing the performance of detectors operating over fading channels, and it may be argued that the subsequent results do not represent practical modem performance, in the absolute sense. Indeed so, and no claim is made to the contrary. However, the objective here is to select a near maximum likelihood detector that is effective in combatting the signal fading experienced over such channels, and within the framework of this objective, perfect estimation of the sampled impulse response of the channel becomes a reasonable, and indeed sensible, assumption.

It was also mentioned in Section 4.4, that the noise components at the input to the detector are slightly correlated, the correlation having been caused by the receiver filter having non-ideal characteristics (i.e., the spectrum of the receiver filter is not flat as in Fig. 4.2.2). It was expected that this would degrade the performance of the detectors. However, 'spot checks' were carried out using uncorrelated noise, and the performance degradation was found to be negligible. It is thus inferred that the correlation introduced by the receiver filter is not sufficient to affect significantly, the performance of the detectors.

In all tests involving error rates, the same fading sequence has been employed, for a given channel. This facilitates proper comparison of the performance of the various detectors when operating over a given channel. It should be noted here that the three fading

sequences employed for the three channels, have been chosen from about 30 different sequences, such that the chosen ones exhibit 'worst case' conditions for the particular channel parameters. The noise and data sequences however, have been varied at each SNR, in order that the performance is not influenced by the choice of a particular sequence. On average, a total of 5 x 10^6 symbols have been involved in the plotting of each curve, making their 95% confidence limits better than ±0.5 dB. The signal-to-noise ratio (SNR), is as defined in eqn. 4.4.17, and the channel parameter values are those given in Table 2.5.2. The number of components in the sampled impulse response of Channels 1, 2 and 3 are, respectively, 22, 22 and 20. The bit error rate is that in the $\{\alpha_i'\}$ (Fig. 4.4.1).

In all the figures showing the performance of the detectors, the nonlinear equalizer is denoted NLE. The near maximum likelihood detectors are named according to the number of expansions per chosen stored vector, and the number of stored vectors. For example, 4B16 is the pseudoquaternary (4 expansions) version of System B with 16 stored vectors, while 2B8 is the pseudobinary (2 expansions) version of System B with 8 stored vectors. System C has no numeral preceding the letter C, because the number of expansions of a stored vector is a variable. All the near maximum likelihood detectors introduce a delay of 32 sampling intervals, i.e., n = 32. The nonlinear equalizer of course, introduces no delay in detection.

Figs. 4.6.1 - 4.6.3 show the fading encountered over the three channels. In terms of the depth of the fades, Channel 1 is the mildest, with the deepest fade around -14dB, and Channel 2 is the severest, with the deepest fade extending to -20dB. This is as expected, since Channel 1 is the sole 3-skywave channel, and the existence of the additional skywave decreases the probability of the signal being in a deep fade at any given time instant. The lower frequency spread (1 Hz.) in Channel 3, is manifested in Fig. 4.6.3 by the slower variations of the signal level. Channel 2 is thus confirmed as the worst of the three channels, exhibiting faster variations and possessing the deepest fades.

Fig. 4.6.4 shows the performance of the system under conditions of no fading, i.e., when the HF radio channel is taken out. The detector used here is 4A16. The curve marked 'ideal' shows the performance when the sampled impulse response of the channel is 1, 0, 0, ... , and the curve marked 'back-to-back' shows the performance when the distortion present is only that caused by the cascade of the transmitter and receiver filters. The corresponding sampled impulse response is given in Table 4.6.1. These curves were obtained in order to assess the degradation in performance caused by the inclusion of the equipment filters, and as can be seen, this degradation is about 0.2 dB.

Before assessing the performance of the various detectors, it is appropriate to

consider first, their relative complexities. Undoubtedly, one of the best measures of complexity is the number of separate instructions needed, for a given digital signal processor (DSP) to implement the algorithms contained in the detectors. To obtain such a measure however, requires quite a thorough understanding of the particular DSP architecture and instruction set, and as such, is beyond the scope of this work. Instead, the complexities are compared here on the basis of the number of costs computed per data symbol, and the number of searches through these costs, again, per data symbol. Table 4.6.2 shows these values for the various systems where the number of stored vectors in each case is p. (Obviously, as the number of stored vectors increase, so does the complexity, making, for example, 4A16 a more complex detector than 4A4). As can be seen from Table 4.6.2, by far the most complex detector is System A. This is followed by System B (quaternary) and finally, by System B (binary) and System C, where the latter two are of a similar order of complexity.

Figs. 4.6.5 - 4.6.8 show the performance of the various detectors when operating over Channel 1. The nonlinear equalizer is the worst of the detectors tested while 4A16 is the best, the difference in performance being about 3 dB asymptotically. As expected, for a given system, the performance improves as the number of stored vectors is increased. An interesting observation here is that this performance improvement is greater when the number of stored vectors is increased from 4 to 8, than when they are increased from 8 to 16, with the latter offering only a very marginal improvement. Thus, on the basis of their performance over Channel 1, it could be said, firstly, that it is more than likely that increasing the number of stored vectors beyond 16 would offer negligible improvement in performance and, secondly, that the detector offering the best compromise between performance and complexity is one which contains 8 stored vectors.

Figs. 4.6.9 - 4.6.12 shows the performance of the detectors when operating over Channel 2, which is the worst of the three channels. As expected, the performance of all the detectors is inferior, relative to their performance over Channel 1. The difference in performance between the nonlinear equalizer and 4A16 (again, the worst and the best of the detectors tested), approaches 2dB, asymptotically. The variation in the performance of a given system with the number of stored vectors, is as observed previously over Channel 1, lending support to the conclusion that a detector with 8 stored vectors should be chosen as that which offers the best compromise between performance and complexity.

Figs. 4.6.13 - 4.6.16 show the performance of the detectors when operating over Channel 3. The difference in performance between the best (4A16) and the worst (nonlinear equalizer) detector here, approaches 4.5dB, and the variation of the performance of the detectors, with the number of stored vectors, is as observed in Channels 1 and 2. It

was thus decided that a detector with 8 stored vectors should be chosen as the preferred system.

Figs. 4.6.17 - 4.6.19 show the performance of the detectors with eight stored vectors (i.e., 4A8, 2B8, 4B8 and C8), when operating over Channels 1, 2 and 3. Since Channel 2 is the worst channel, the initial choice of a detector would be based on its performance over this channel, the reasoning being that a detector which gives satisfactory performance over poor channels, should perform well over channels with less adverse fading conditions. However, Fig. 4.6.18 shows that there is at most, only 1dB difference in performance between the four detectors, when operating over Channel 2. It could also be seen that, although the best detector here is 4A8, the detector C8 exhibits a similar performance to the former. Since C8 is also much less complex than 4A8, it is the most suitable detector, given the objectives stated earlier. Moreover, Figs. 4.6.17 and 4.6.19 show that the relative performance of these detectors over Channels 1 and 3 is similar to their relative performance over Channel 2, thus confirming that C8 should be chosen as the detector which gives the best compromise between performance and complexity.

The performance of C8 over the 3 channels is summarised in Fig. 4.6.20. As expected, the best performance is over Channel 1 and the worst performance over Channel 2, the degradation being about 6dB, asymptotically. It should be noted here, that the degradation at high signal-to-noise ratios is greater than that at low signal-to-noise ratios. This is due to the increasing influence at high SNR's, of the deep fades, on the performance of the system. Most of the errors at high SNR's are caused by deep fades, and since Channel 2 has deeper fades than Channel 1, the degradation in performance of a system operating over Channel 2 would become more pronounced as the signal-to-noise ratio increases. Thus, the relative asymptotic behaviour of these curves is probably a more reliable indicator of the vagaries introduced by the channels, than is their behaviour at low SNR's. The asymptotic performance of C8 over Channel 3 falls somewhere between that of its performance over Channels 1 and 2.

References - Chapter 4

1. **Clark, A.P.,** 'Advanced data transmission systems', Pentech Press, 1977.

2. **Clark, A.P.,** 'Equalizers for digital modems', Pentech Press, 1985.

3. **Chang, R.W. and Hancock, J.C.,** 'On receiver structures for channels having memory', IEEE Trans. on Information Theory, Vol. IT-12, pp. 463-468, 1966.

4. **Forney, G.D.,** 'Maximum likelihood sequence estimation in the presence of intersymbol interference', IEEE Trans. on Information Theory, Vol. IT-18, pp. 363-378, May 1972.

5. **Viterbi, A.J.,** 'Error bounds for convolutional codes and an asympotically optimum decoding algorithm', IEEE Trans. on Information Theory, Vol. IT-13, pp. 260-269, April 1967.

6. **Forney, G.D.,** 'The Viterbi algorithm', IEEE Proceedings, Vol. 61, No. 3, pp. 268-278, March 1973.

7. **Najdi, H.,** 'Digital data transmission over voice frequency channels', Ph.D thesis, Loughborough University of Technology, 1982.

8. **Thomas, J.B.,** 'An introduction to statistical communication theory', John Wiley & Sons Inc., 1969.

9. **Magee, F.R., and Proakis, J.G.,** 'An estimate of the upper bound on error probability for maximum likelihood sequence estimation on channels having a finite duration pulse response', IEEE Trans. on Information Theory, Vol. IT-19, pp. 699-702, Sept. 1973.

10. **Ungerboeck, G.,** 'Adaptive maximum likelihood receiver for carrier modulated data transmission systems, IEEE Trans. on Communications, Vol. COM-22, No. 5, pp. 624-636, May 1974.

11. **Anderson, I.N.,** 'Sample-whitened matched filters', IEEE Trans. on Information Theory, Vol. IT-19, pp.120-124, 1973.

12. **Clark, A.P. and Fairfield, M.J.,** 'Detection processes for a 9600 bit/s modem', Radio and Electronic Engineer, 51, pp.455-465, 1981.

13. **Qureshi, S.U.H., and Newhall, E.E.,** 'An adaptive receiver for data transmission over time dispersive channels, IEEE Trans. on Information Theory, Vol. IT-19, pp.448-457, July 1973.

14. **Falconer, D.D., and Magee, F.R.,** 'Adaptive channel memory truncation for maximum likelihood sequence estimation, Bell Syst. Tech. Journal, Vol .52, pp.1541- 1562, Nov. 1973.

15. **Desblache', A.E.,** 'Optimal short desired impulse response for maximum likelihood sequence estimation, IEEE Trans. on Communications, Vol. COM-25, No. 7, pp.735-738, July 1977.

16. **Beare, C.T.,** 'The choice of the desired impulse response in combined linear - Viterbi algorithm equalizers, IEEE Trans. on Communications, Vol. COM-26, No. 8, pp.1301- 1307, August 1978.

17 **Lee, W.U., and Hill, F.S.,** 'A maximum likelihood sequence estimator with decision feedback equalization, IEEE Trans. on Communications, Vol. COM-25, No. 9, pp.971-979, Sept. 1977.

18. **Vermeuler, F.L., and Hellman, M.E.,** 'Reduced-state Viterbi decoders for channels with intersymbol interference', Conf. record, IEEE Int. Conf. on Communications, 17-19 June, 1974, Minneapolis, Minnesota, pp. 32.B.1-4, 1974.

19. **Foschini, G.J.,** 'A reduced-state variant of maximum likelihood sequence detection attaining optimum performance for high signal-to-noise ratios, IEEE Trans. on Information Theory, Vol. IT-23, No. 5, pp.605-609, Sept. 1977.

20. **Clark, A.P., Harvey, J.D. and Driscoll, J.P.** 'Near maximum likelihood detection processes for distorted digital signals', Radio and Electronic Engineer, 48, pp.301- 309, June 1978.

21. **Clark, A.P., and Asghar, S.M.,** 'Detection of digital signals transmitted over a

known time-varying channel', IEE Proc., Part F, 128, pp.167-174, June 1981.

22. **Clark, A.P., and Najdi, H.** 'Detection process of a 9600 bit/s serial modem for HF radio links, IEE Proc., Part F, Vol. 130, No. 5, August 1983.

23. **Clark, A.P., Ip, S.F.A., and Soon, C.W.,** 'Psuedobinary detection processes for a 9600 bit/s modem, IEE Proc., Part F, Vol. 129, No. 5, pp. 305-314, October 1982.

24. **Clark, A.P., and Najdi, H.,** 'Detection process of a 9600 bit/s serial modem for HF radio links', IEE Proc., Part F, Vol. 130, No. 5, pp. 368-376, August 1983.

25. **Clark, A.P., and Clayden, M.,** 'Psuedobinary Viterbi detector', IEE Proc., Part F, Vol. 131, No. 2, pp. 208-218, April 1984.

26. **Clark, A.P., Abdullah, S.N., Jayasinghe, S.G. and Sun, K.H.** 'Psuedobinary and psuedoquaternary detection processes for linearly distorted multilevel QAM signals', IEEE Trans. on Communications, Vol. COM-33, No. 7, pp. 639-645, July 1985.

27. **Clark, A.P. and Abdullah, S.N.,** 'Near maximum likelihood detectors for voiceband channels', IEE Proc., Part F, Vol. 134, No. 3, pp. 217-226, June 1987.

28. **Clark, A.P., and Hau, S.F.,** 'Adaptive adjustment of receiver for distorted digital signals', IEE Proc., Part F, Vol. 131, No. 5, pp. 526-536, August 1984.

29. **Abdullah, S.N.** 'Data transmission at 9600 bit/s over an HF radio link', Ph.D. thesis, Loughborough University of Technology, 1986.

Sampled impulse response of transmitter filter ($\tau = 0$)		Sampled impulse response of receiver filter	
Real Part	Imaginary Part	Real Part	Imaginary Part
–0.1796	2.3539	–1.9418	1.3626
–3.0074	20.7590	–15.9798	11.5941
–9.9409	45.5585	–35.1418	27.3343
–11.7870	41.4910	–34.4789	28.0870
–3.4618	8.7046	–11.2302	7.2715
4.4438	–11.7870	7.8155	–9.2602
3.0643	–5.5819	7.5124	–5.0954
–1.3597	3.1582	–0.5058	3.2327
–1.4974	1.7365	–3.3707	1.8975
0.2926	–0.7777	–0.6760	–1.2817
0.5181	–0.1293	1.0483	–0.4830
–0.1843	0.2880	0.3622	0.7615
–0.3168	–0.2325	–0.3106	0.1979
0.0022	–0.2108	0.0438	–0.1533
–0.0444	0.0392	0.0739	0.0940
0.0516	0.0099	–0.0647	–0.0312

Table 4.4.1 Sampled impulse responses of transmitter and receiver filters, both minimum phase, obtained at 4800 samples/s.

Sampled impulse response of transmitter filter τ = 1.1 ms		Sampled impulse response of transmitter filter τ = 2.0 ms		Sampled impulse response of transmitter filter τ = 3.0 ms	
Real Part	Imaginary Part	Real Part	Imaginary Part	Real Part	Imaginary Part
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	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	0.0000	0.0000	0.0000	0.0000
−1.6694	13.2373	0.0000	0.0000	0.0000	0.0000
−7.8492	39.6494	0.0000	0.0000	0.0000	0.0000
−12.3887	46.9272	0.0000	0.0000	0.0000	0.0000
−6.6023	19.2347	0.0000	0.0000	0.0000	0.0000
2.9409	−8.8804	−0.7630	7.3452	0.0000	0.0000
4.3005	−9.0256	−5.6487	31.9050	0.0000	0.0000
−0.3368	1.6284	−11.9216	48.7718	0.0000	0.0000
−1.9014	2.8139	−9.3589	29.8080	0.0000	0.0000
−0.1434	−0.4311	0.5650	−3.0208	0.0000	0.0000
0.6243	−0.4537	4.9376	−11.4980	−1.3137	11.0689
0.0279	0.3082	1.0473	−0.9823	−7.1104	37.2137
−0.3820	−0.0772	−1.9766	3.5053	−12.3470	47.9575
−0.0417	−0.3043	−0.7165	0.3116	−7.5849	22.8263
−0.0440	0.0085	0.5944	−0.7219	2.2354	−7.2499
0.0749	0.0094	0.2544	0.2045	4.5939	−10.0027
−0.0594	0.0095	−0.3636	0.1085	0.0932	0.8695
		−0.1544	−0.3287	−1.9704	3.1073
		−0.0228	−0.0636	−0.3234	−0.2261
		0.0167	0.0279	0.6313	−0.5553
		−0.0610	0.0186	0.1036	0.2882
				−0.3866	−0.0157
				−0.0735	−0.3216
				−0.0387	−0.0108
				0.0608	0.0141
				−0.0710	0.0136

Table 4.4.2 Sampled impulse responses of the transmitter filter for different delays, obtained at 4800 samples/s., and shown relative to the transmitter filter with no delay

Sampled impulse response	
Real Part	Imaginary Part
1.00000	0.00000
0.45076	0.06770
−0.16580	−0.03527
0.35403	0.01254
0.00629	0.00069
−0.00415	−0.00209
0.00005	0.00117
−0.00011	−0.00008
0.00021	−0.00005
−0.00011	0.00000
0.00005	0.00002
0.00000	0.00000
0.00000	0.00000
0.00000	0.00000
0.00000	0.00000
0.00000	0.00000

Table 4.6.1 **Sampled impulse response of cascade of transmitter ($\tau = 0$) and receiver filter, obtained at 2400 samples/s.**

	Number of costs per data symbol	Number of searches through the costs
System A	4 p	p
System B (binary)	1.5 p	0.5 p
System B (quaternary)	1.75 p	0.25 p
System C	1.5 p	variable

p = number of stored vectors

Table 4.6.2 **Relative complexities of detectors**

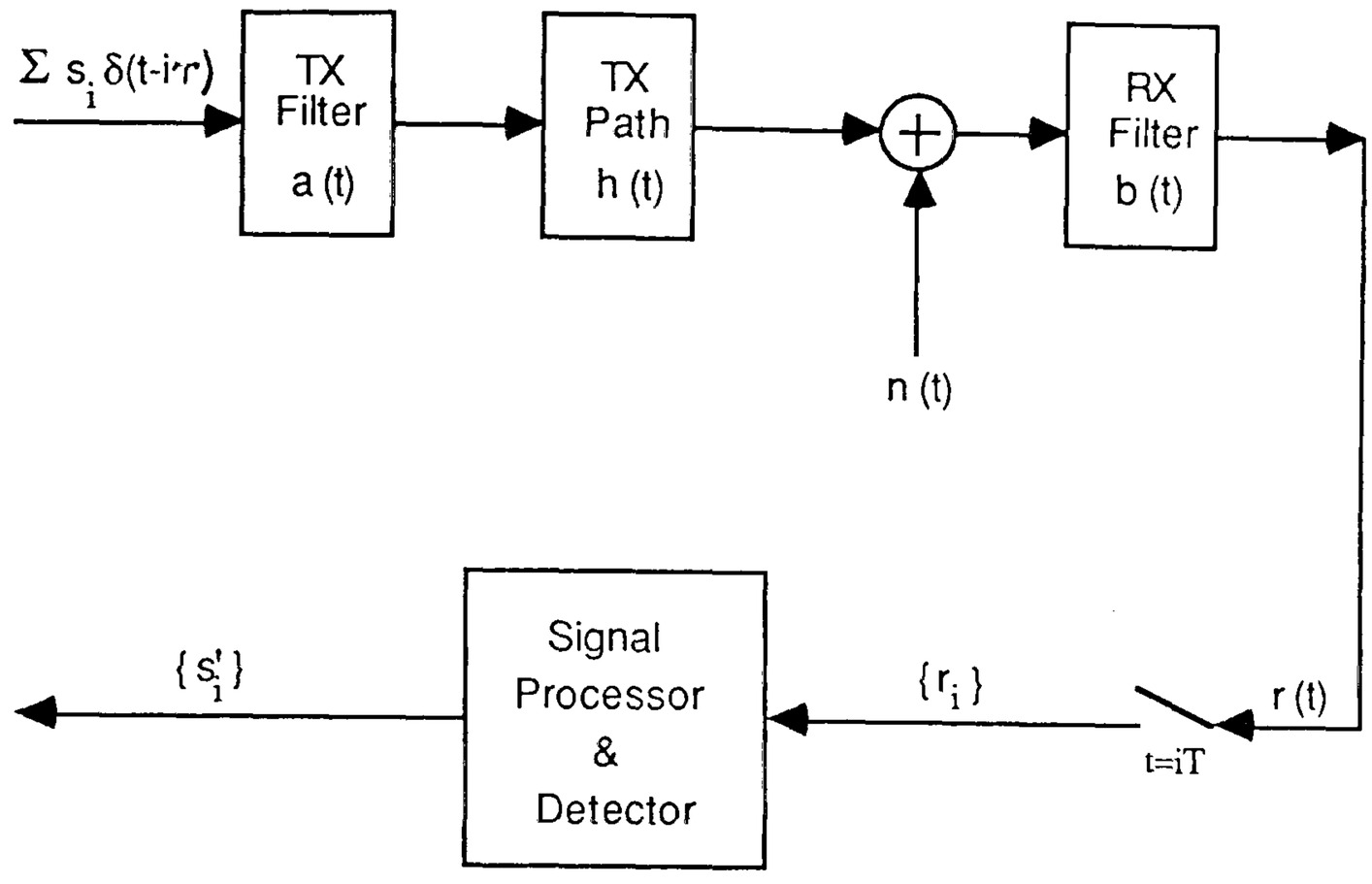


Fig. 4.2.1 **Model of synchronous, serial data transmission system**

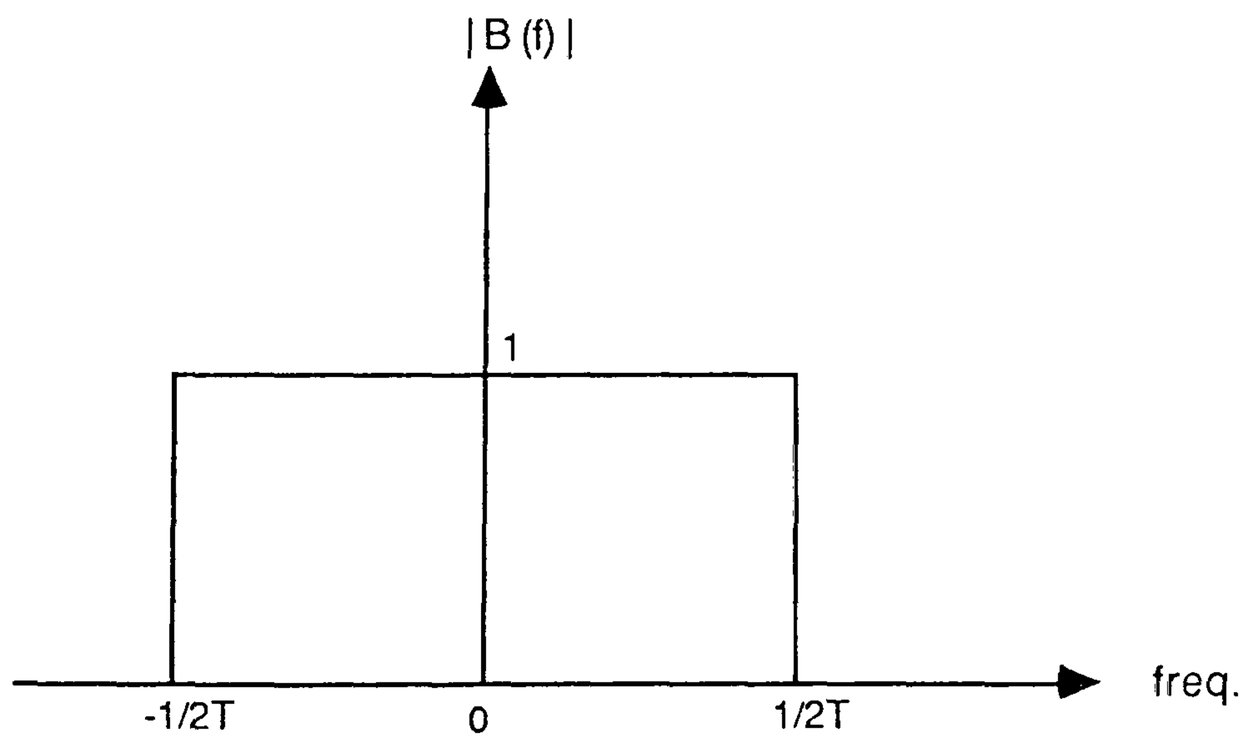


Fig. 4.2.2 **Absolute value of transfer function of receiver filter**

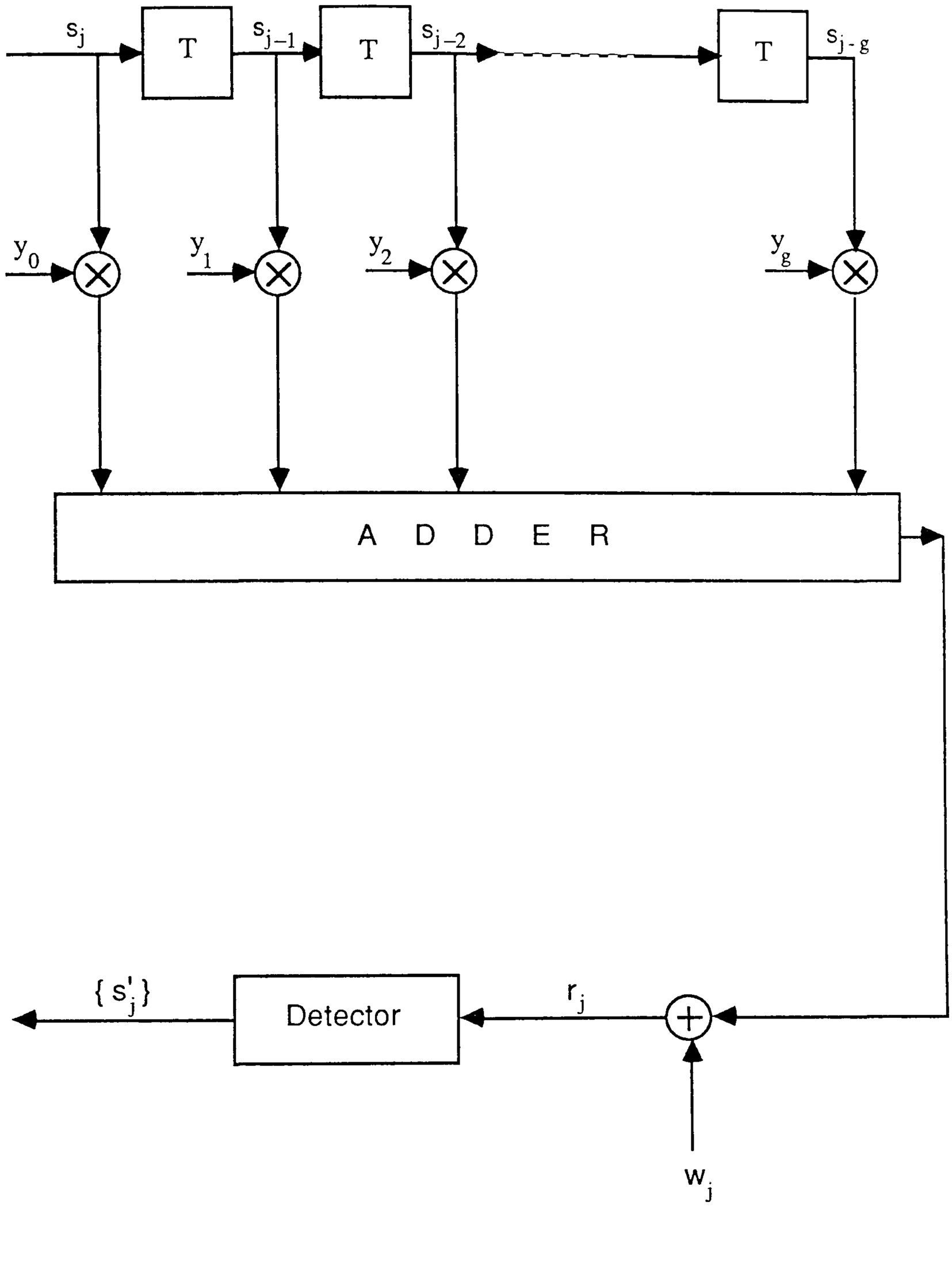


Fig. 4.2.3 Equivalent shift register model of data transmission system

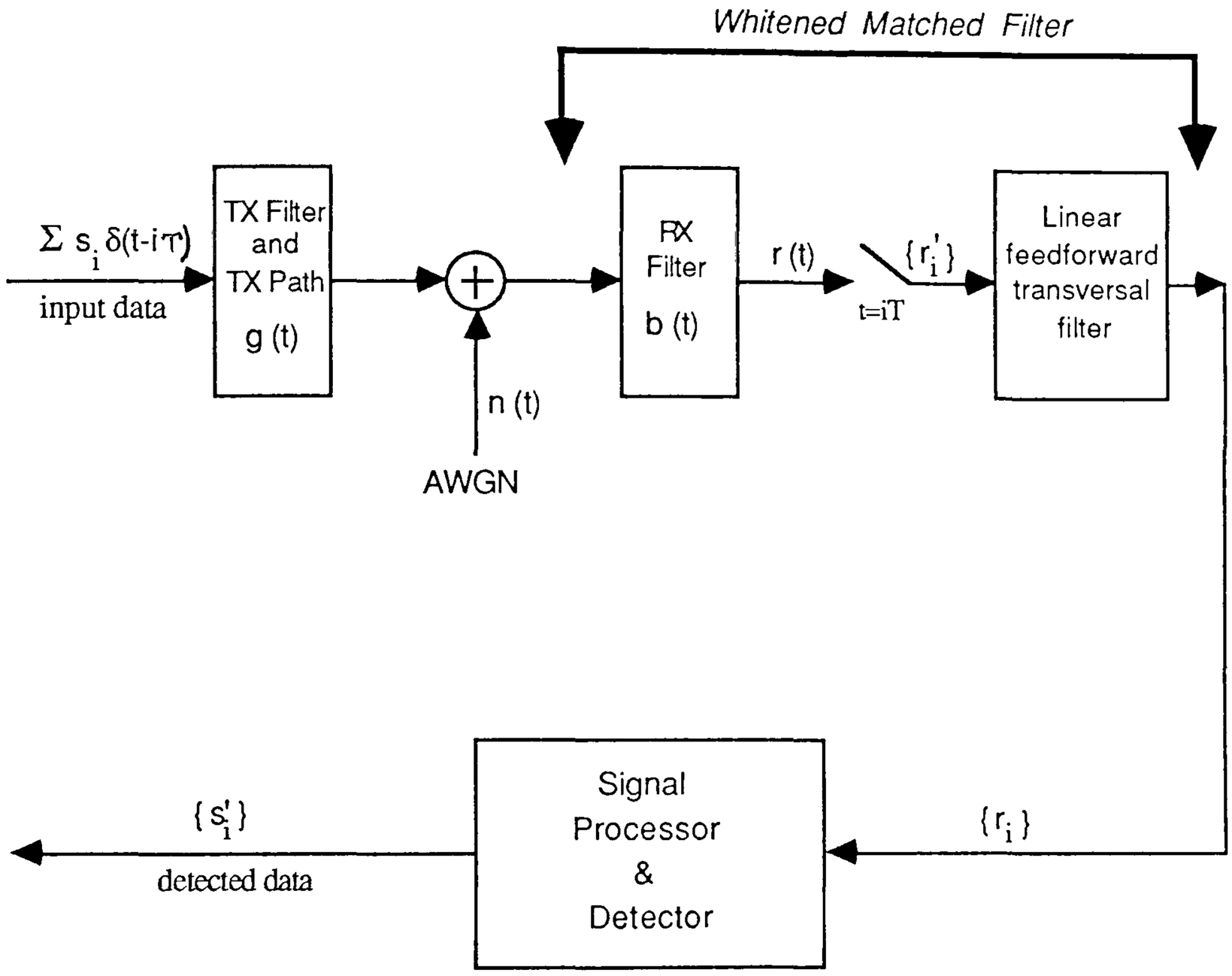


Fig. 4.3.1 **Model of data transmission system using a whitened matched filter**

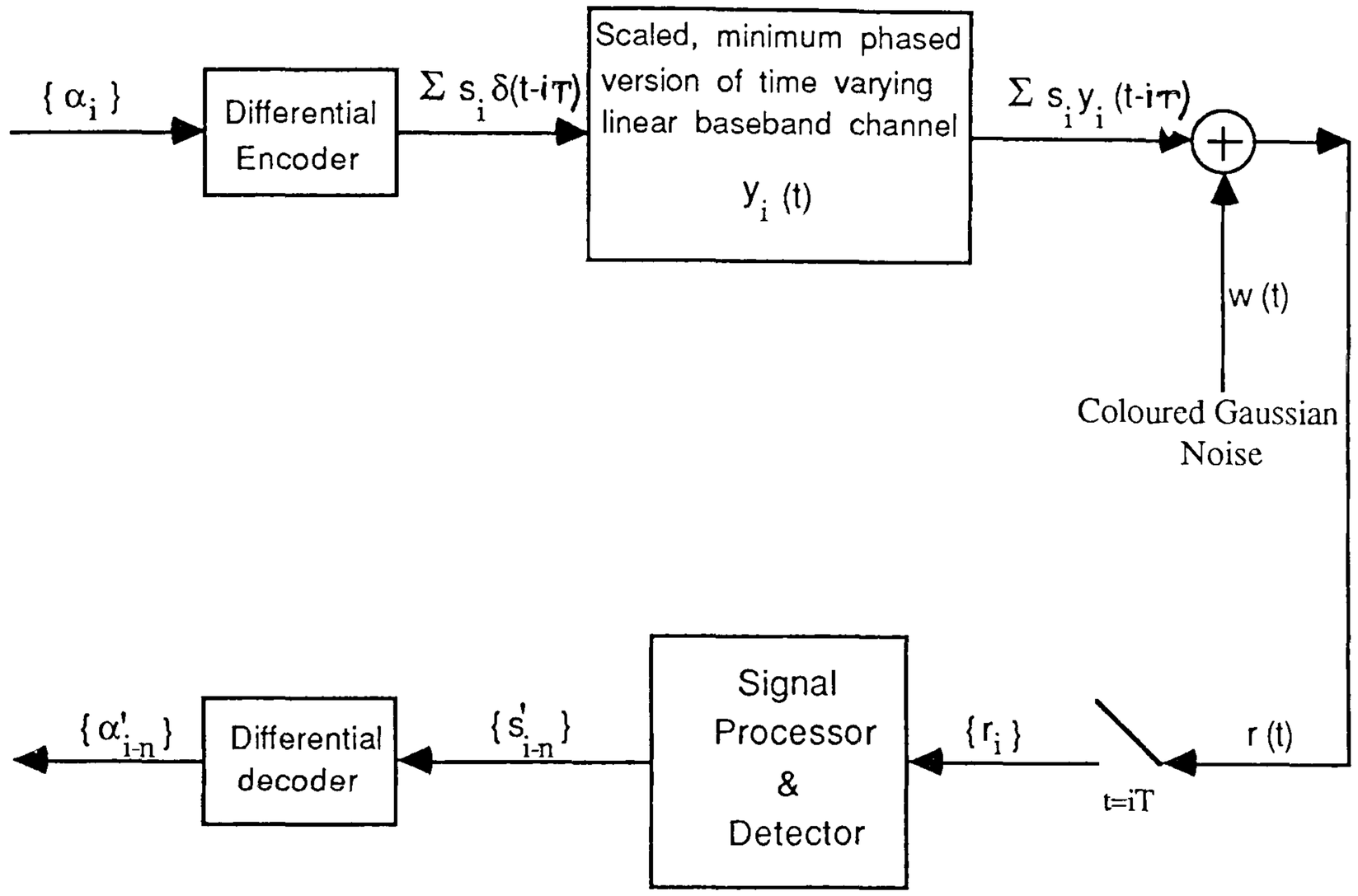


Fig. 4.4.1 Model of data transmission system used in the simulation

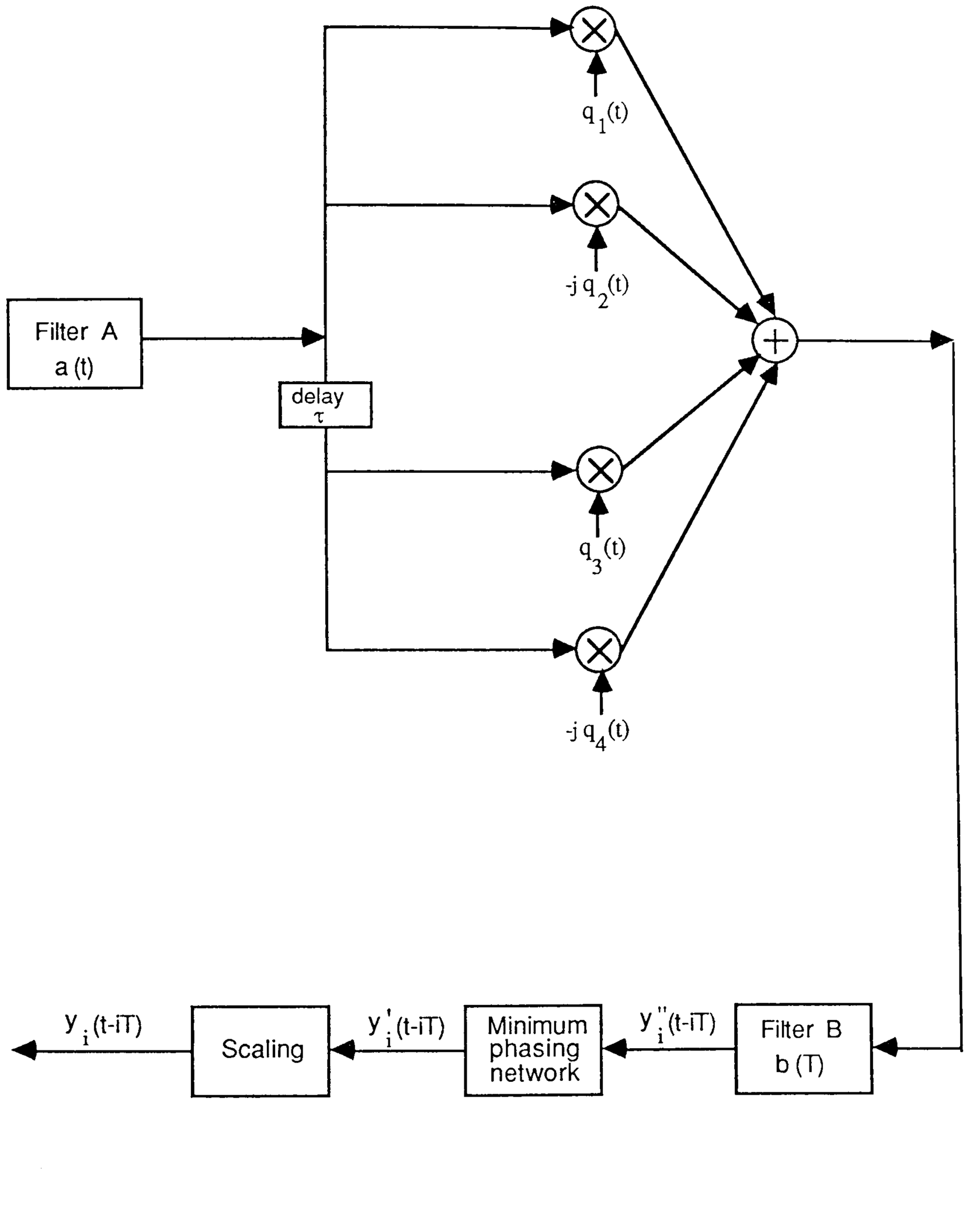


Fig. 4.4.2 Scaled, minimum phased, linear baseband channel model

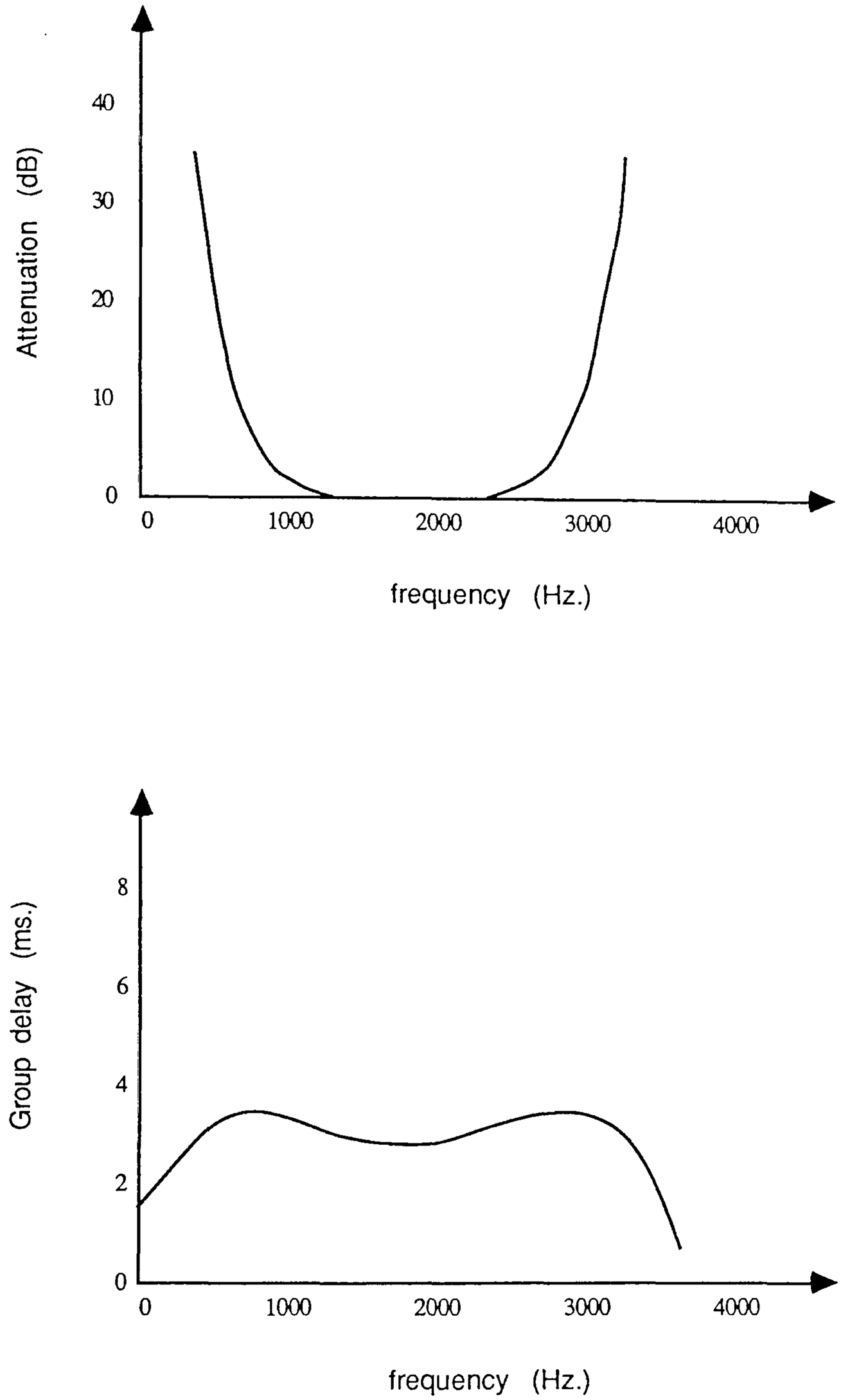


Fig. 4.4.3 Characteristics of the modem transmitter and receiver filters in cascade, in the passband of the QAM signal

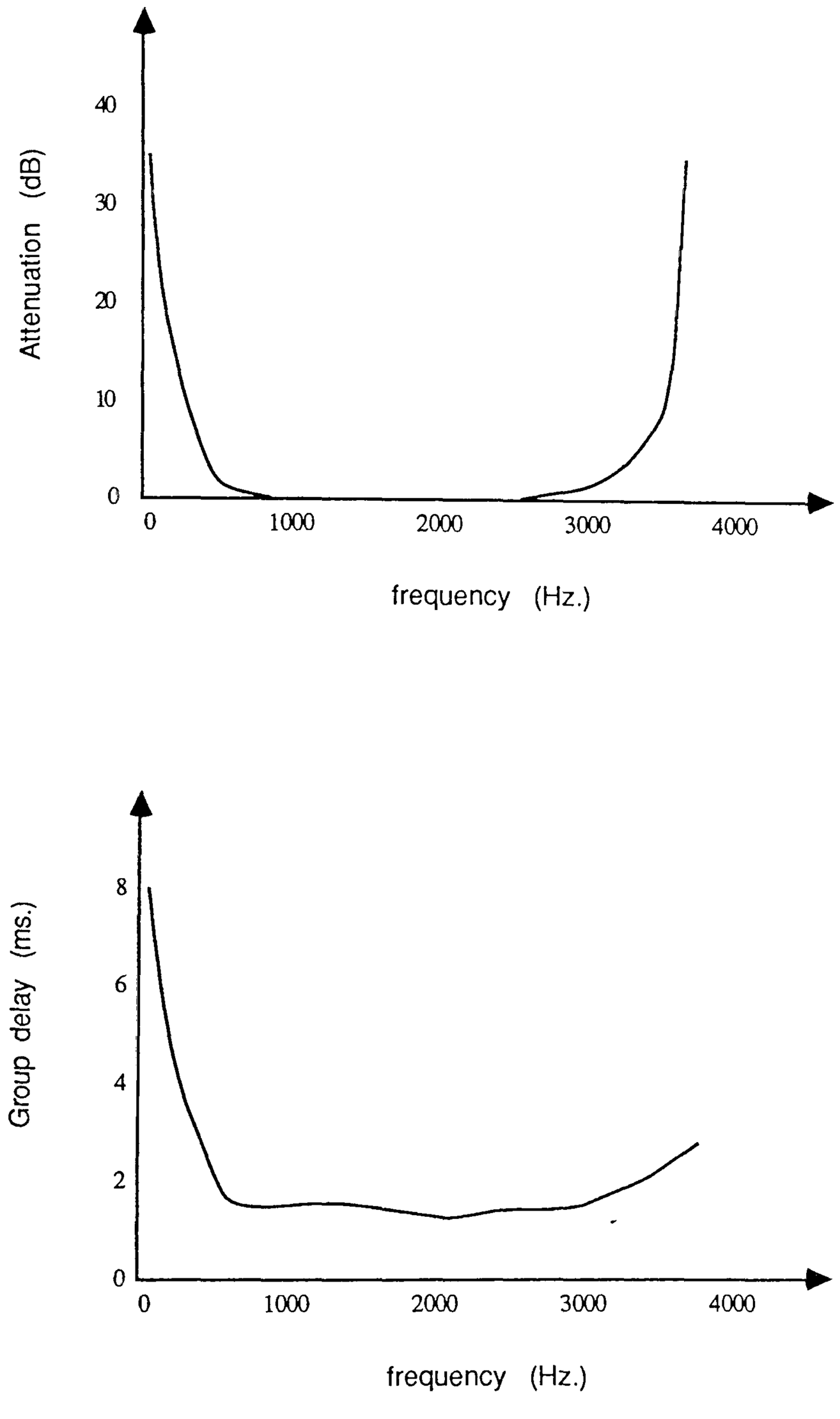


Fig. 4.4.4 **Characteristics of the Clansman VRC 321 radio filters**

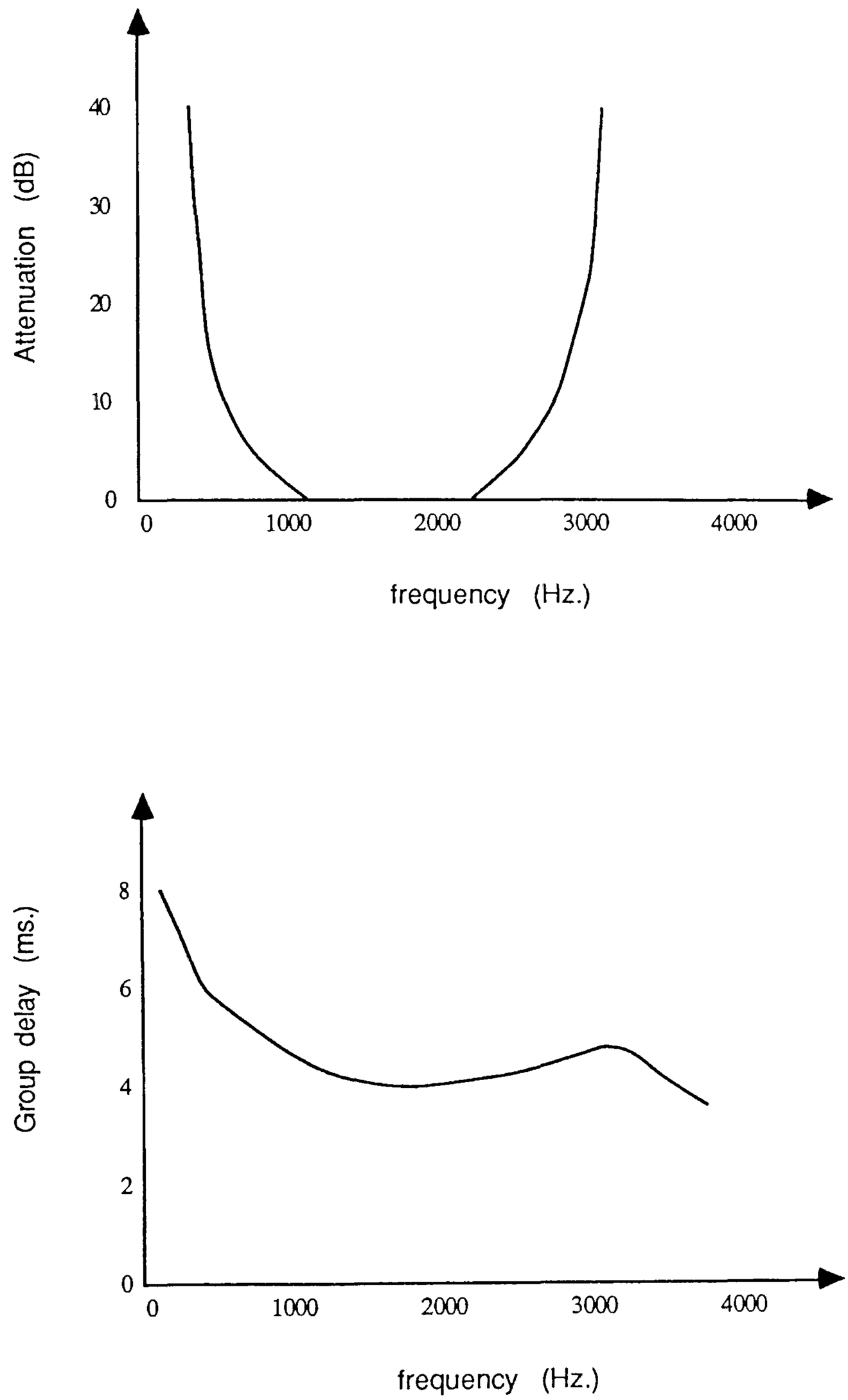


Fig. 4.4.5 Characteristics of the transmitter and receiver filters of the modem and radio equipment in cascade

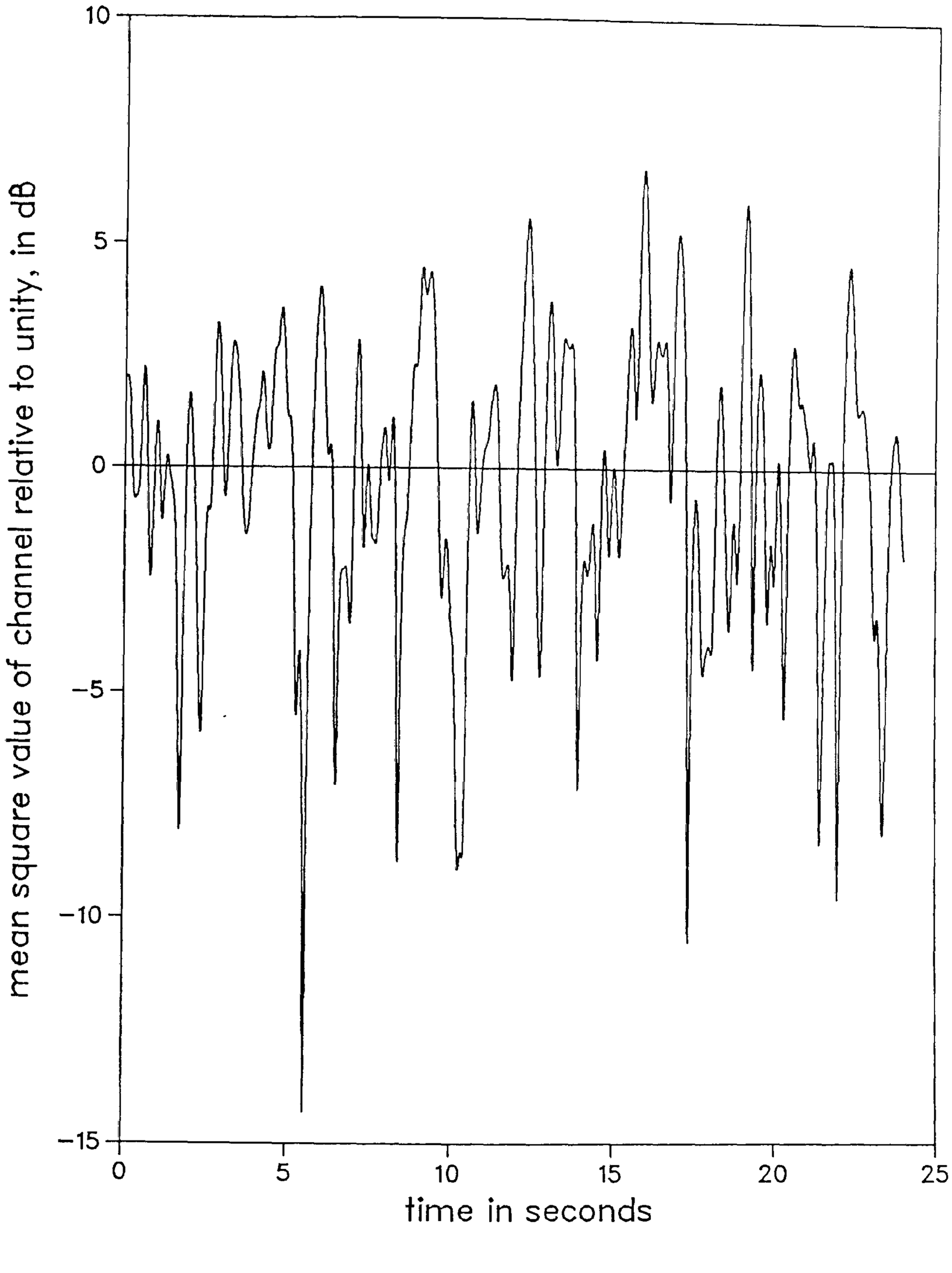


Fig.4.6.1. Fading in Channel 1

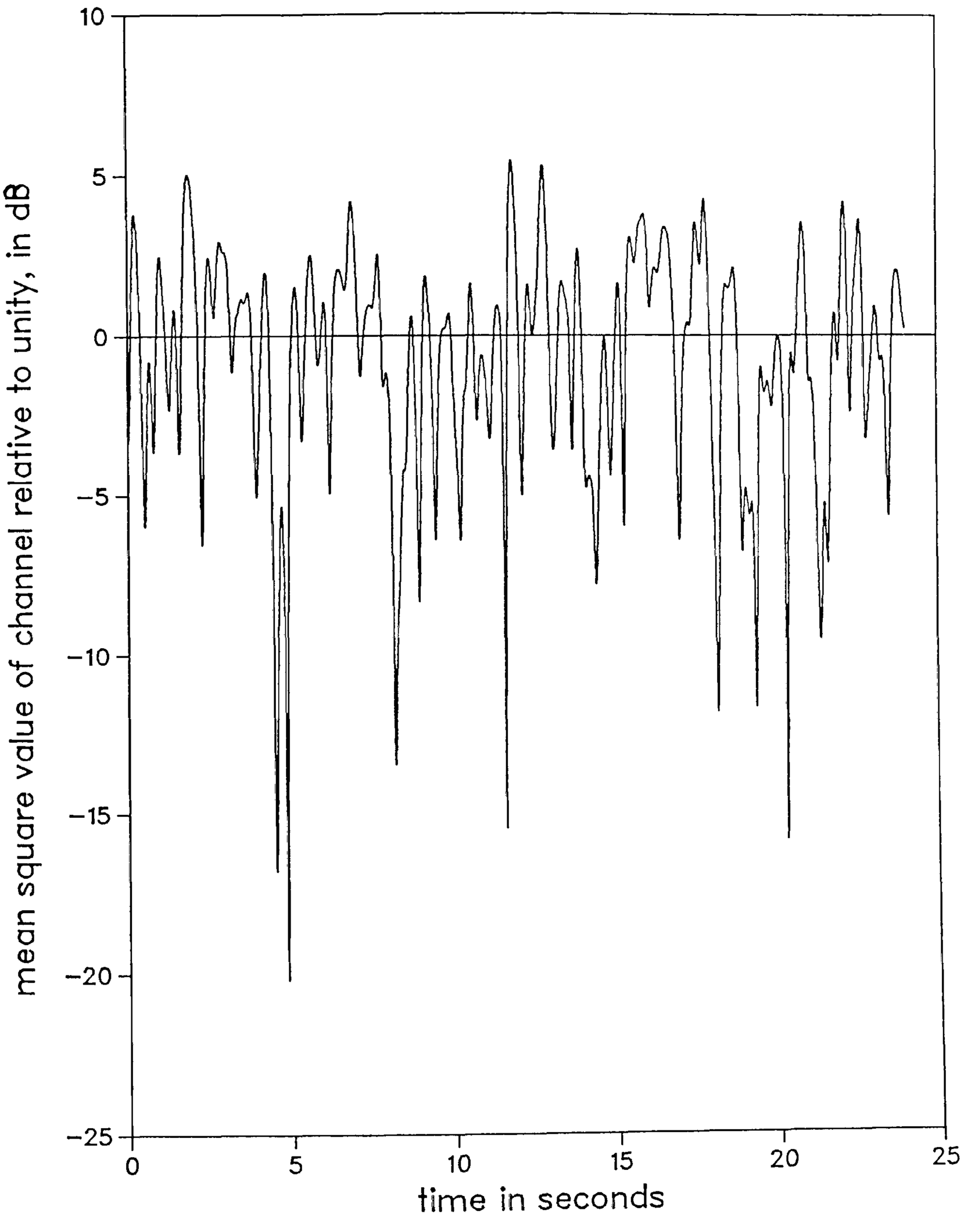


Fig.4.6.2. Fading in Channel 2

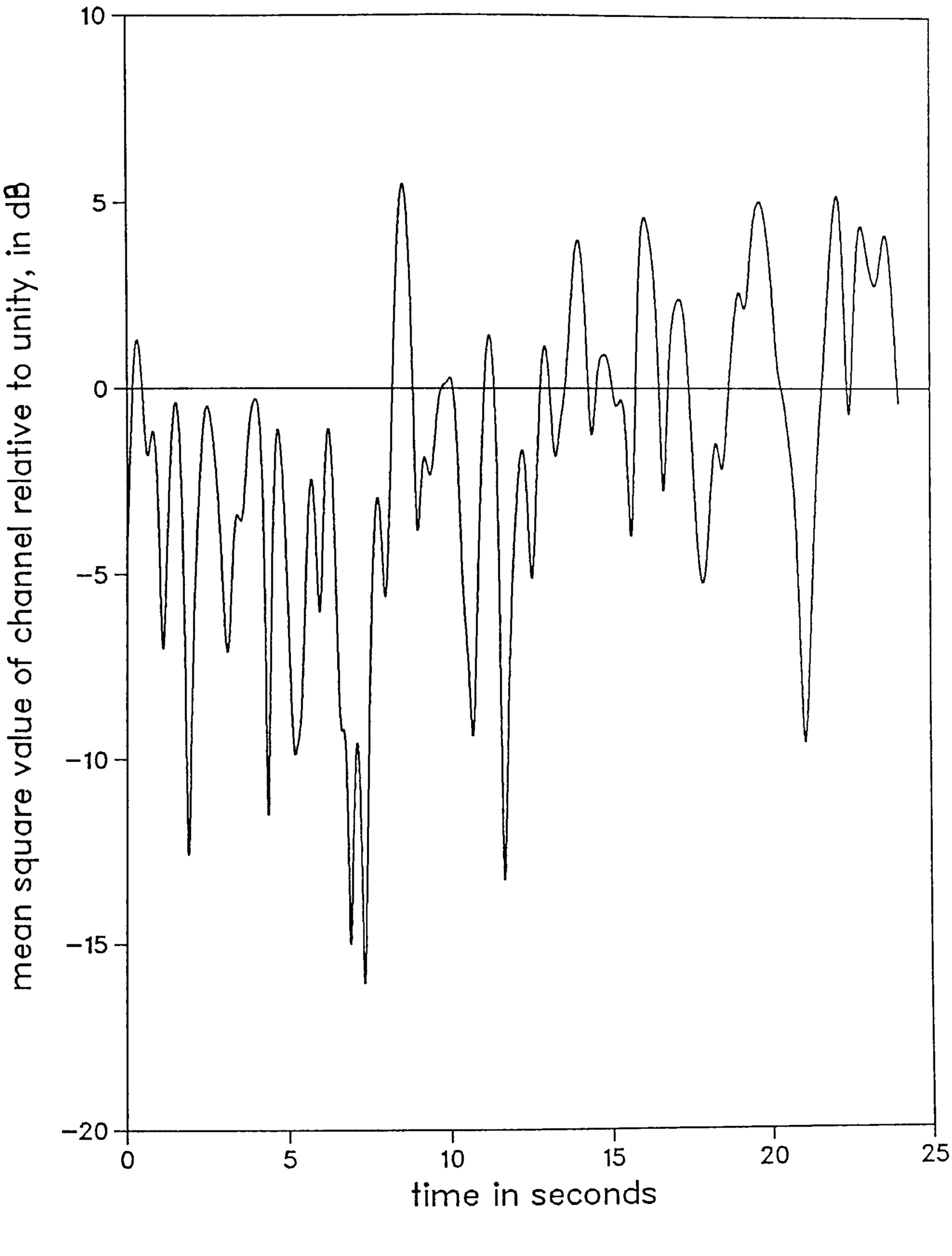


Fig.4.6.3. Fading in Channel 3

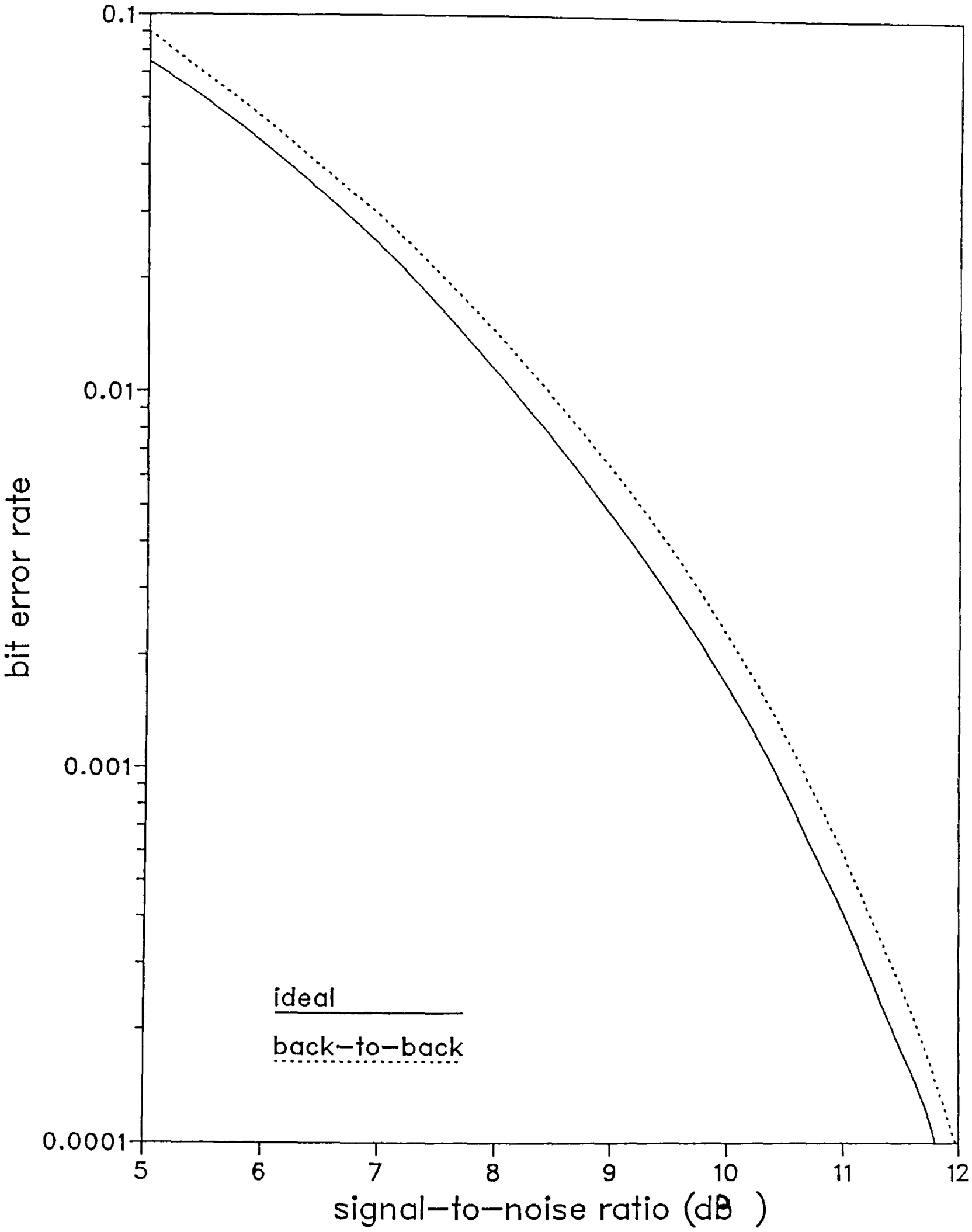


Fig.4.6.4. Performance of 4A16 in the absence of fading

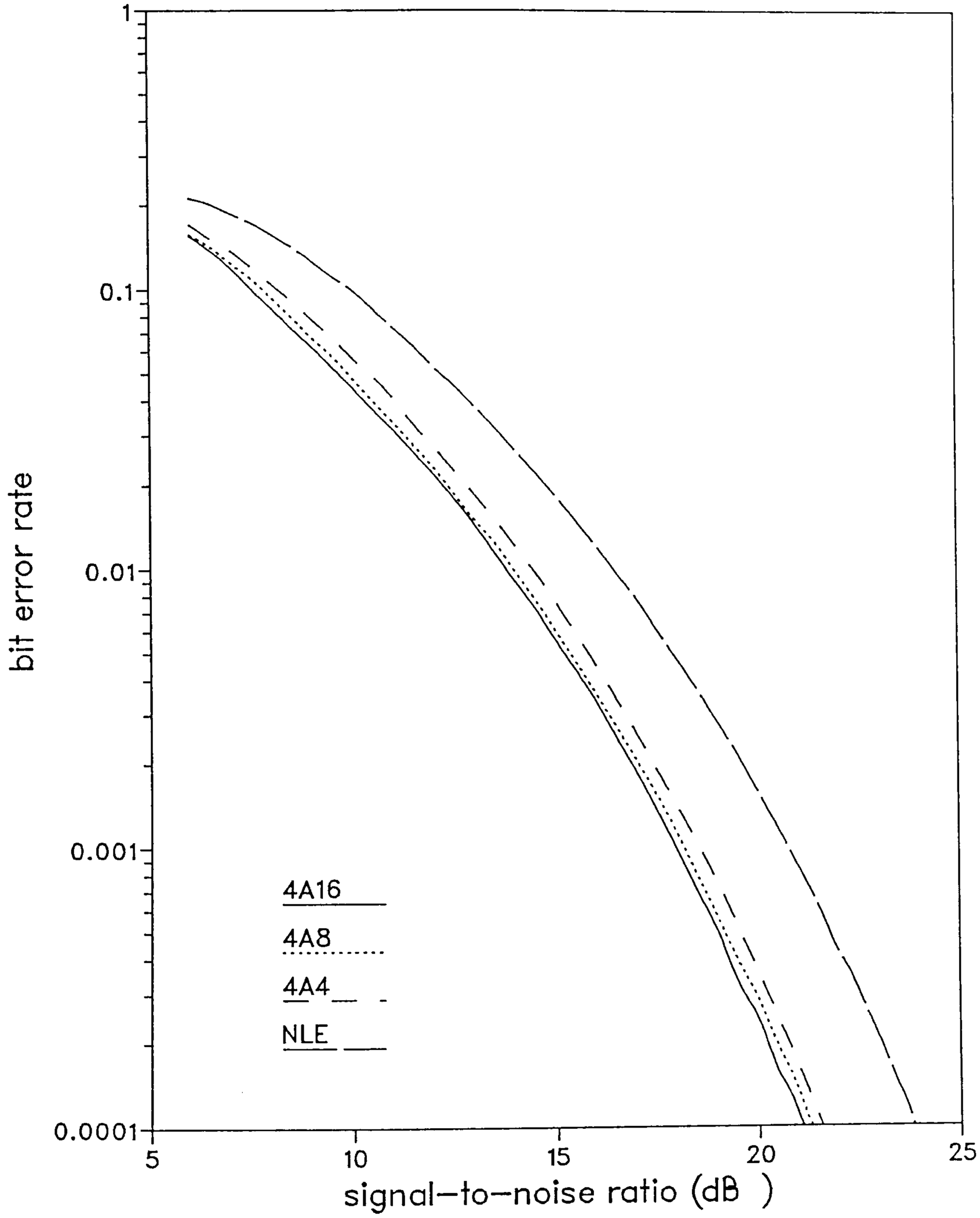


Fig.4.6.5. System A operating over Channel 1

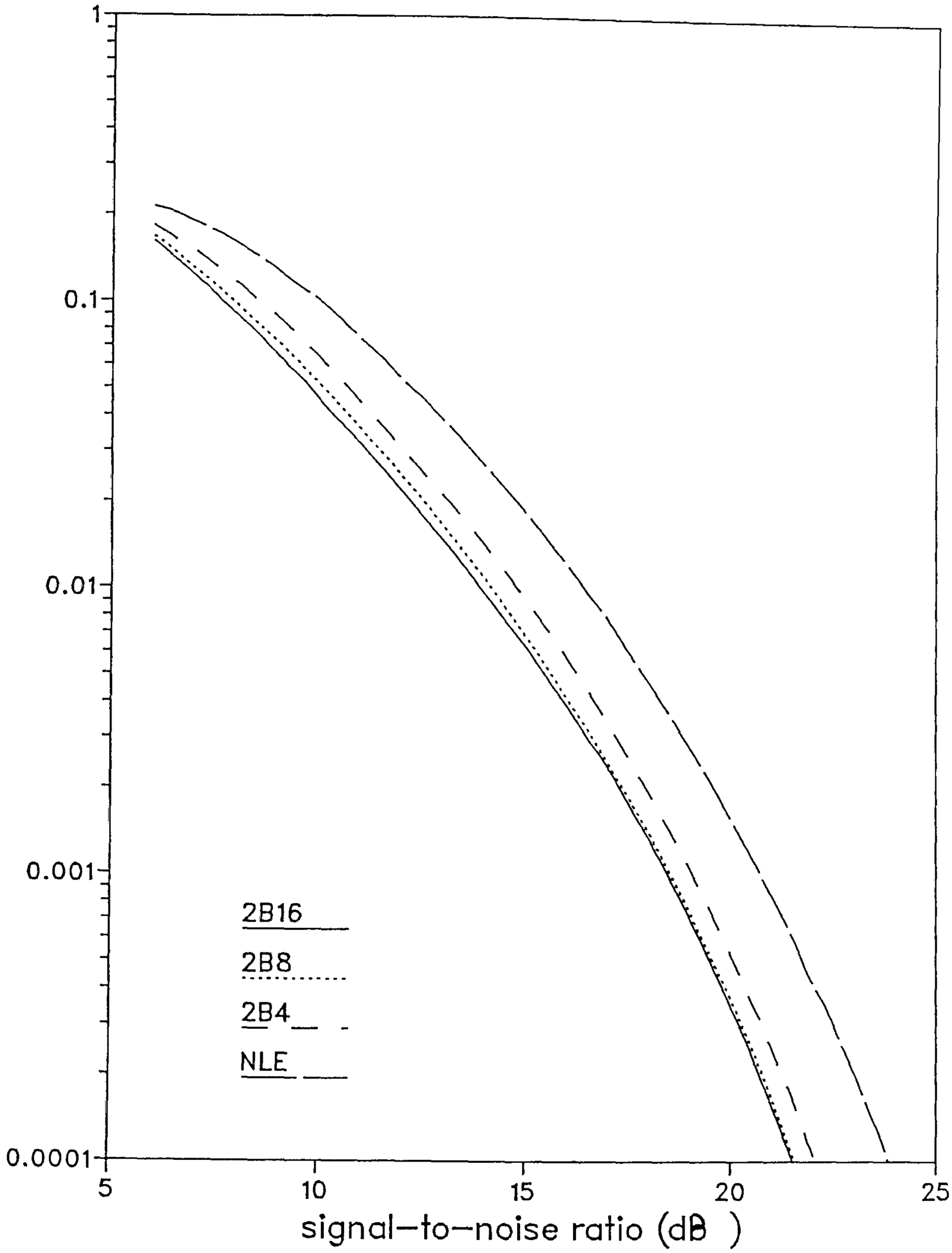


Fig.4.6.6. System B (binary) operating over Channel 1

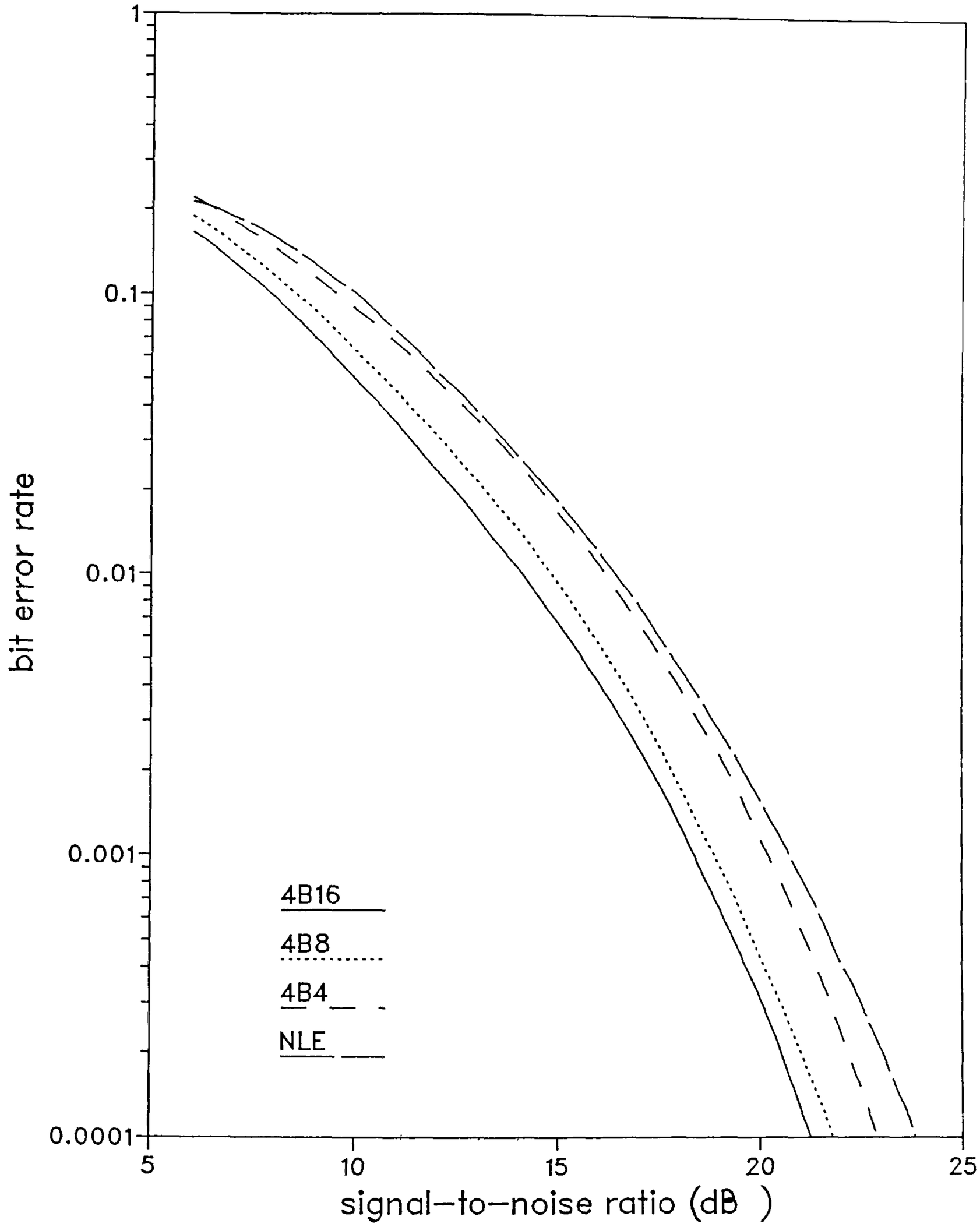


Fig.4.6.7. System B (quaternary) operating over Channel 1

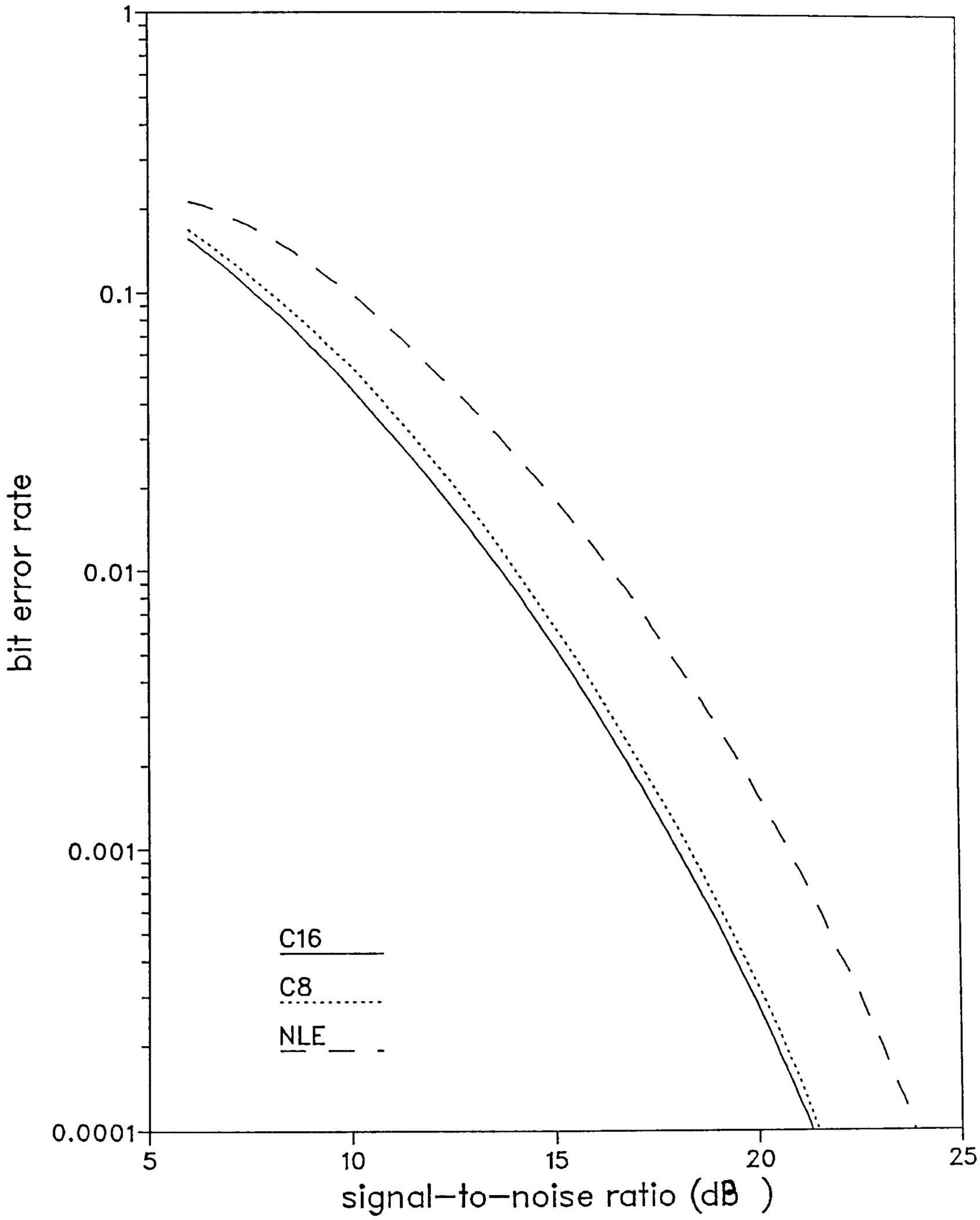


Fig.4.6.8. System C operating over Channel 1

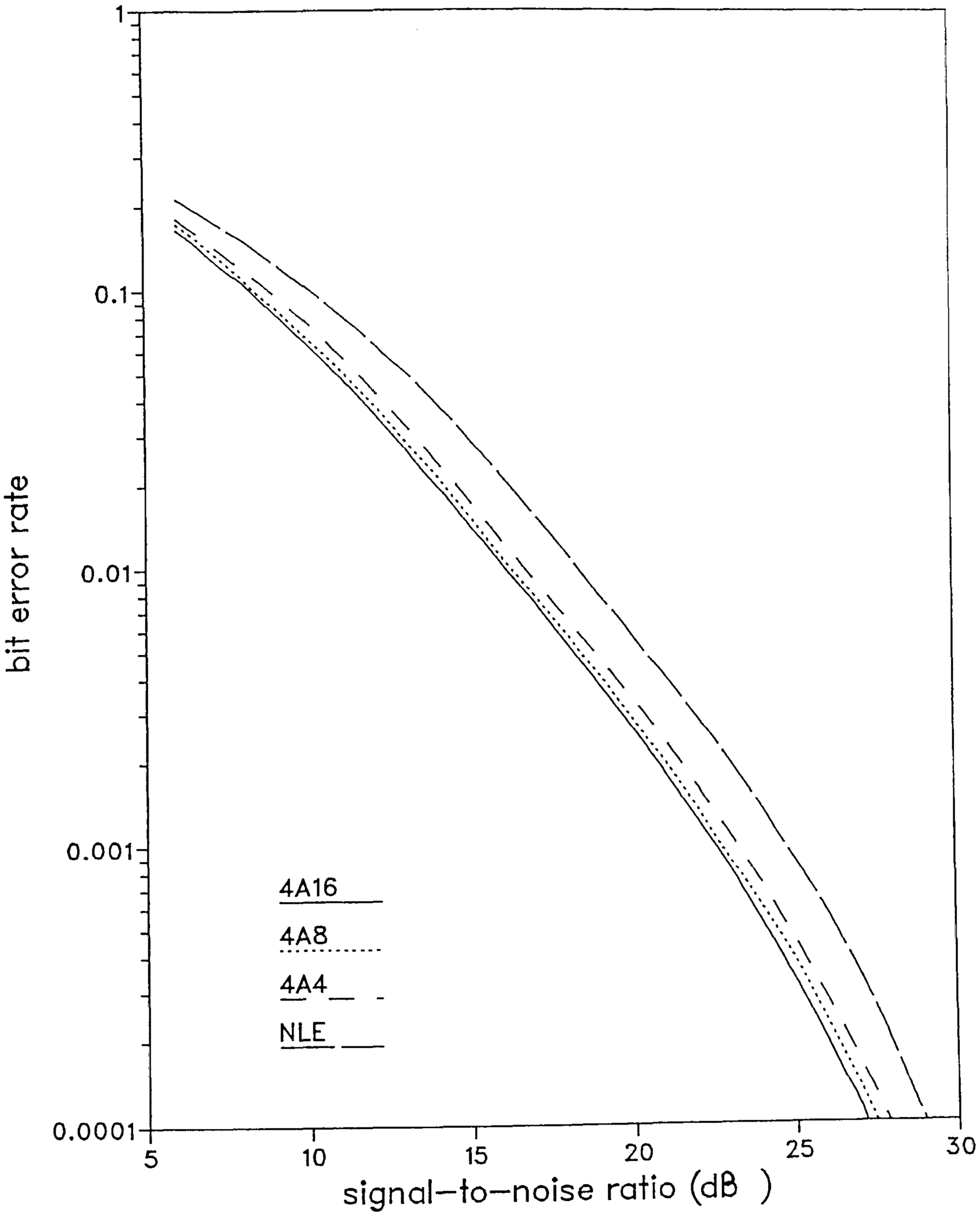


Fig.4.6.9. System A operating over Channel 2

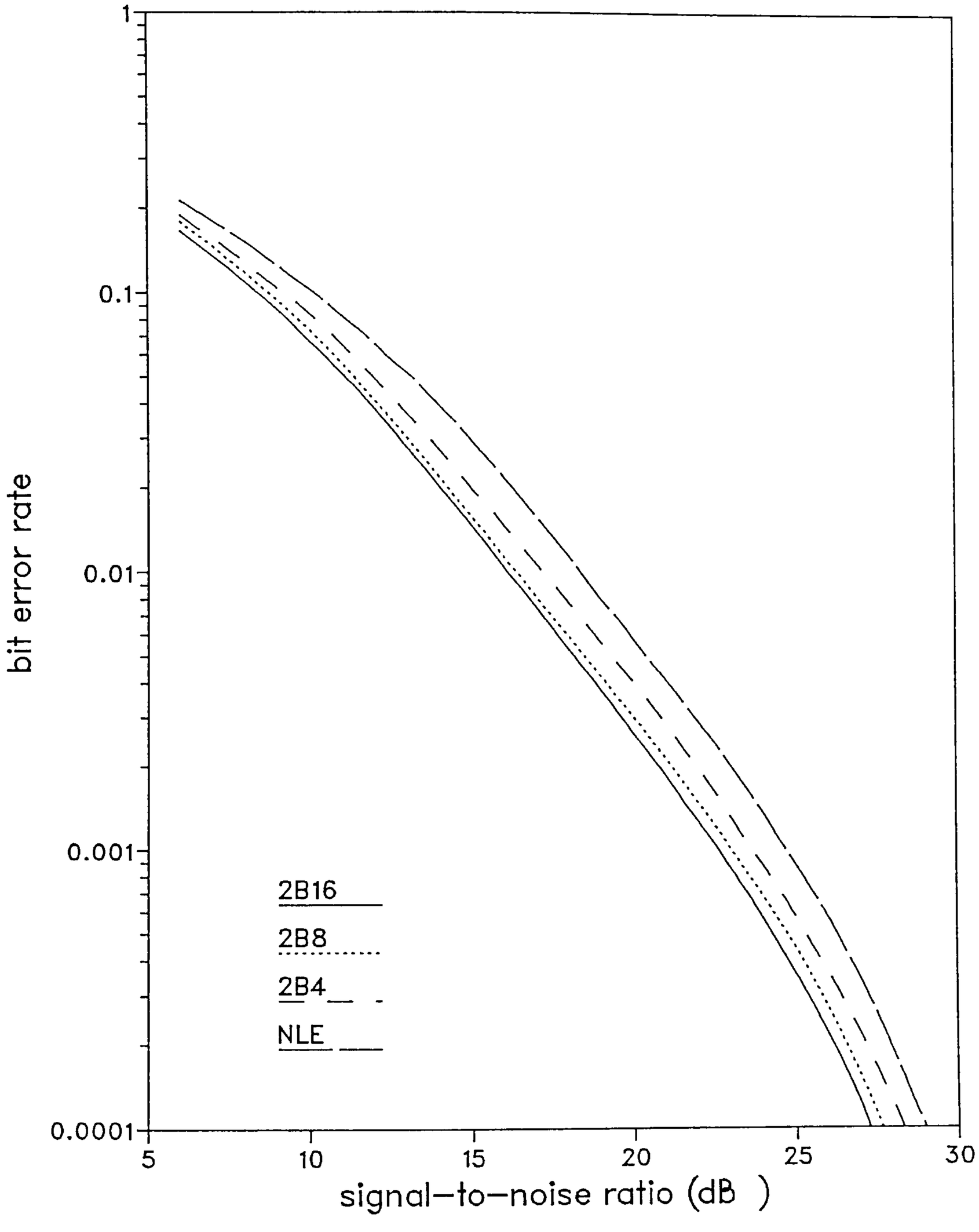


Fig.4.6.10. System B (binary) operating over Channel 2

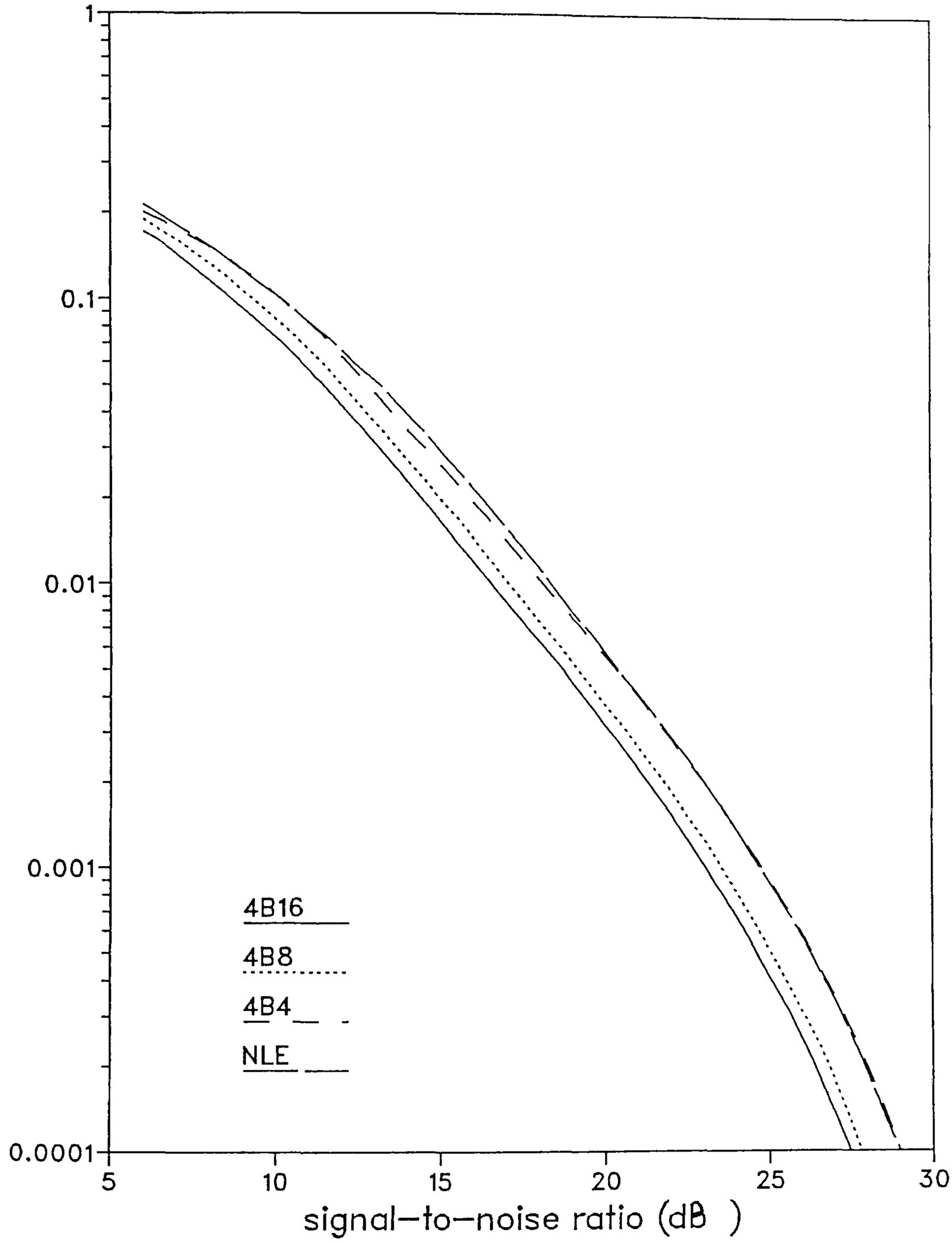


Fig.4.6.11. System B (quaternary) operating over Channel 2

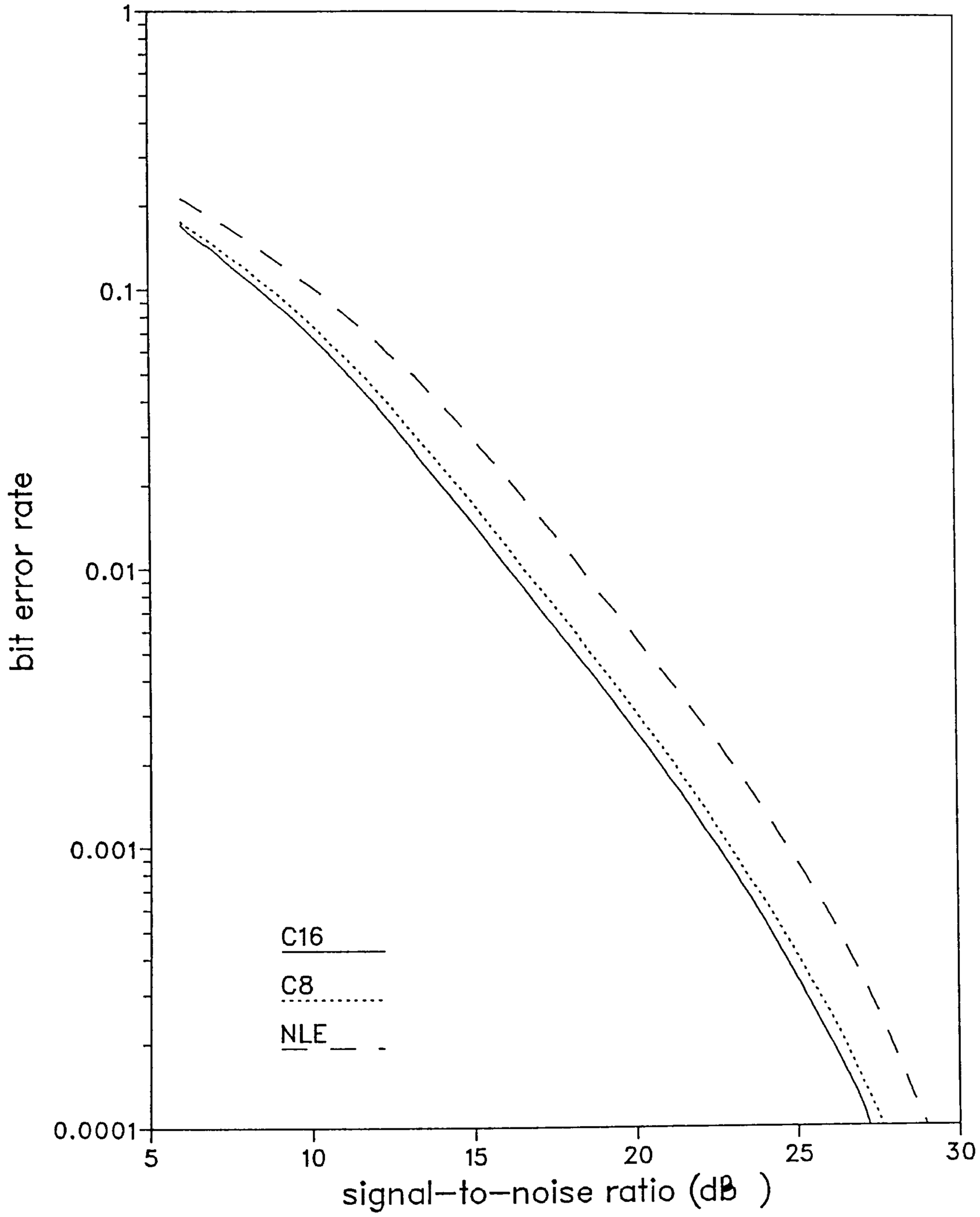


Fig.4.6.12. System C operating over Channel 2

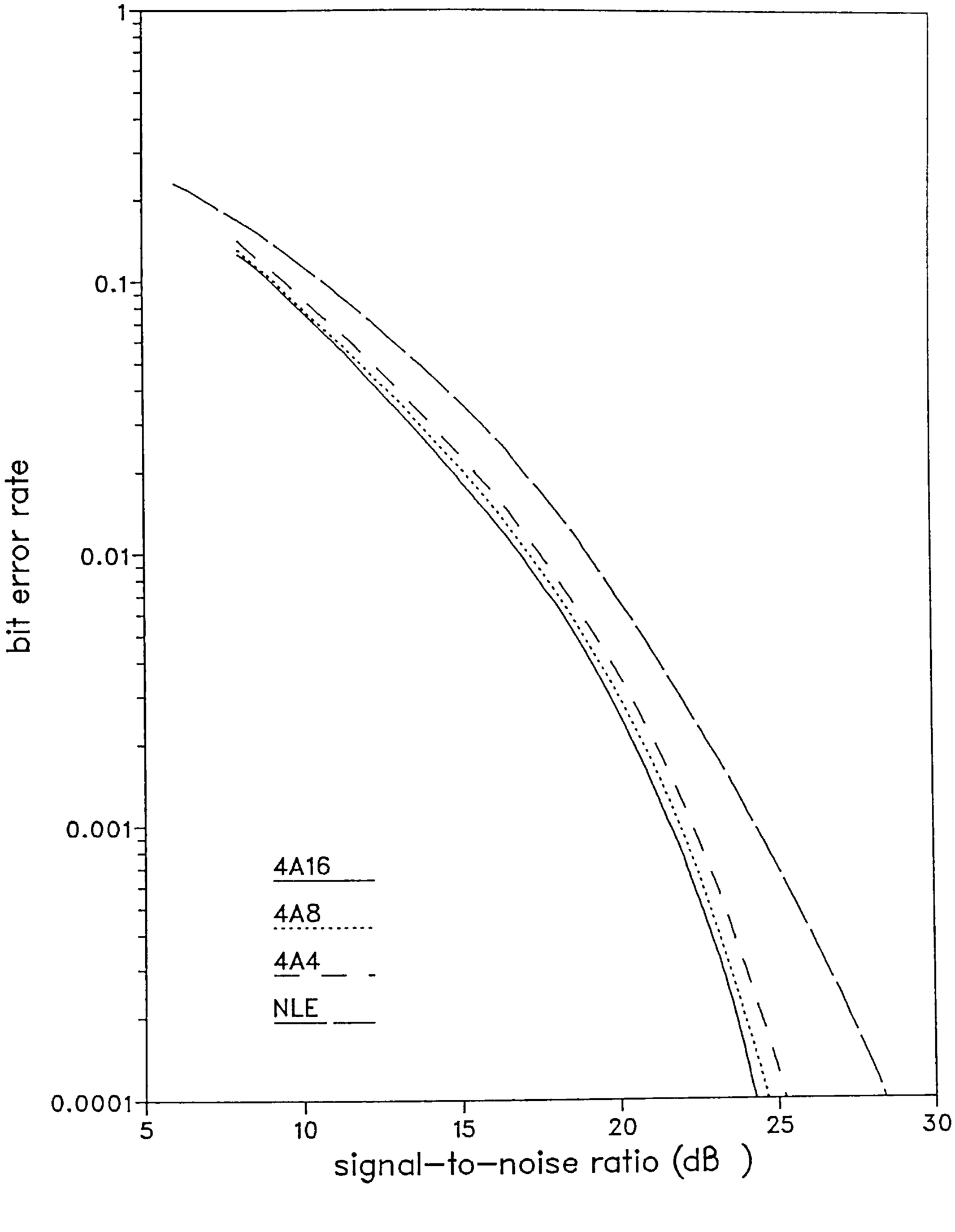


Fig.4.6.13. System A operating over Channel 3

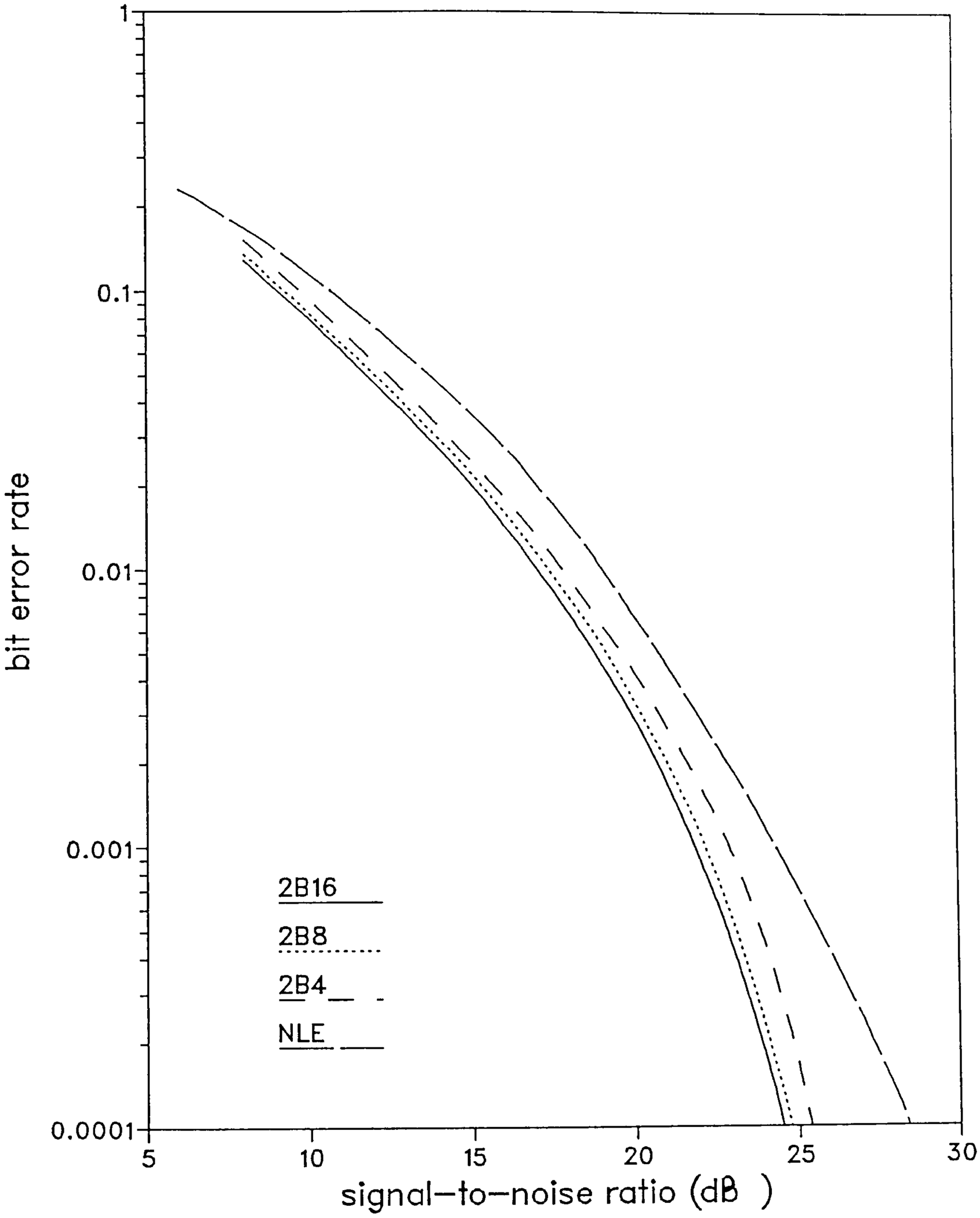


Fig.4.6.14. System B (binary) operating over Channel 3

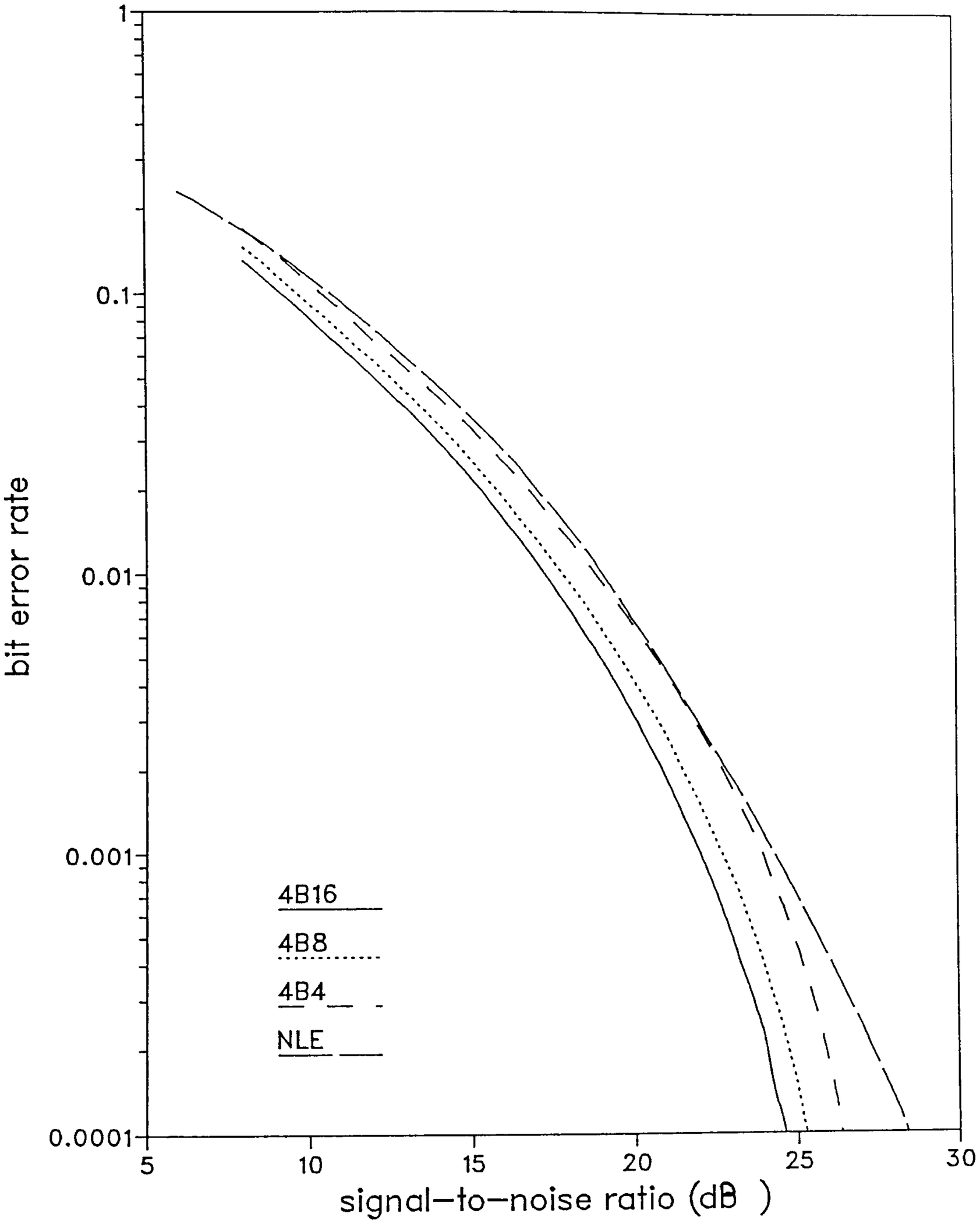


Fig.4.6.15. System B (quaternary) operating over Channel 3

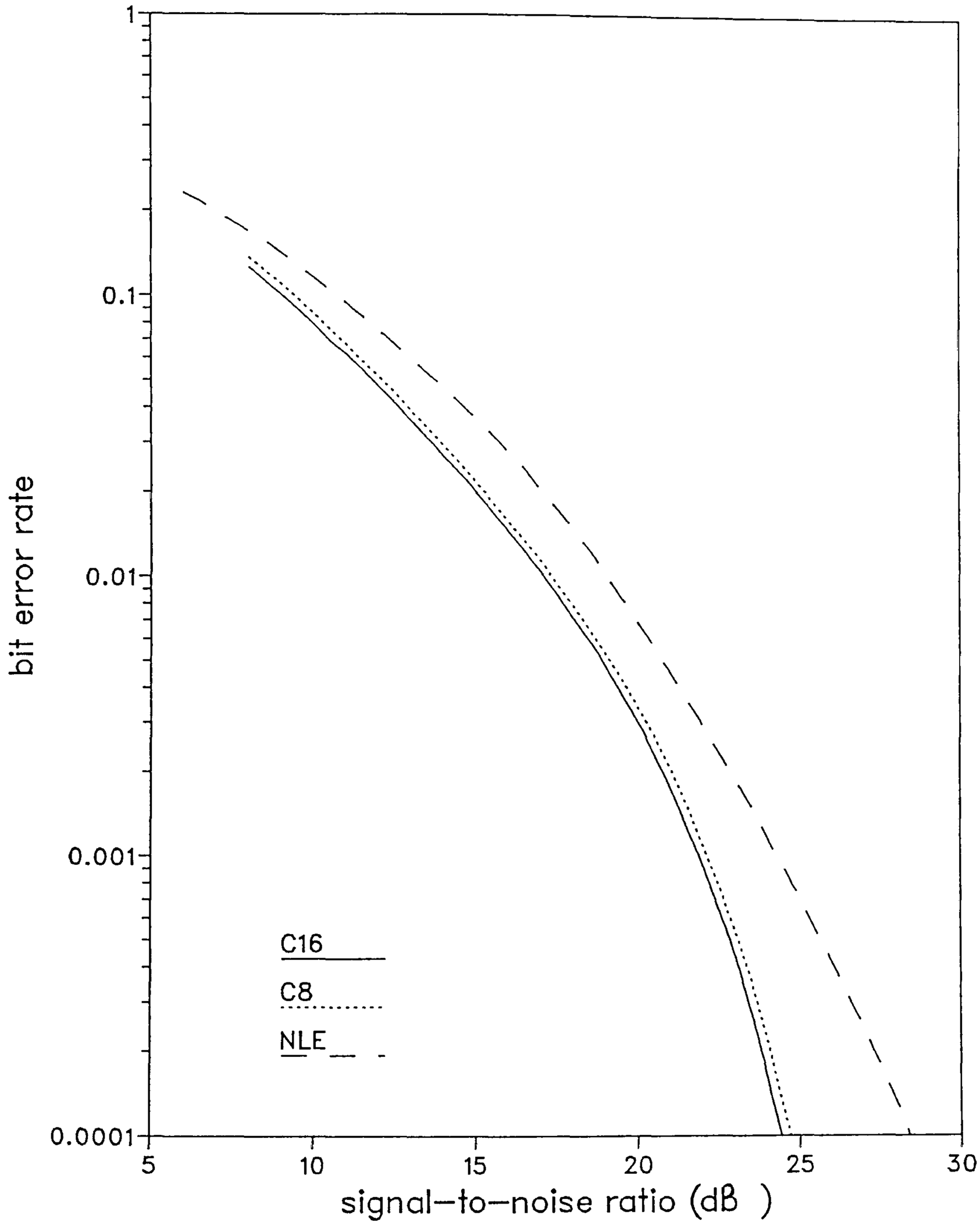


Fig.4.6.16. System C operating over Channel 3

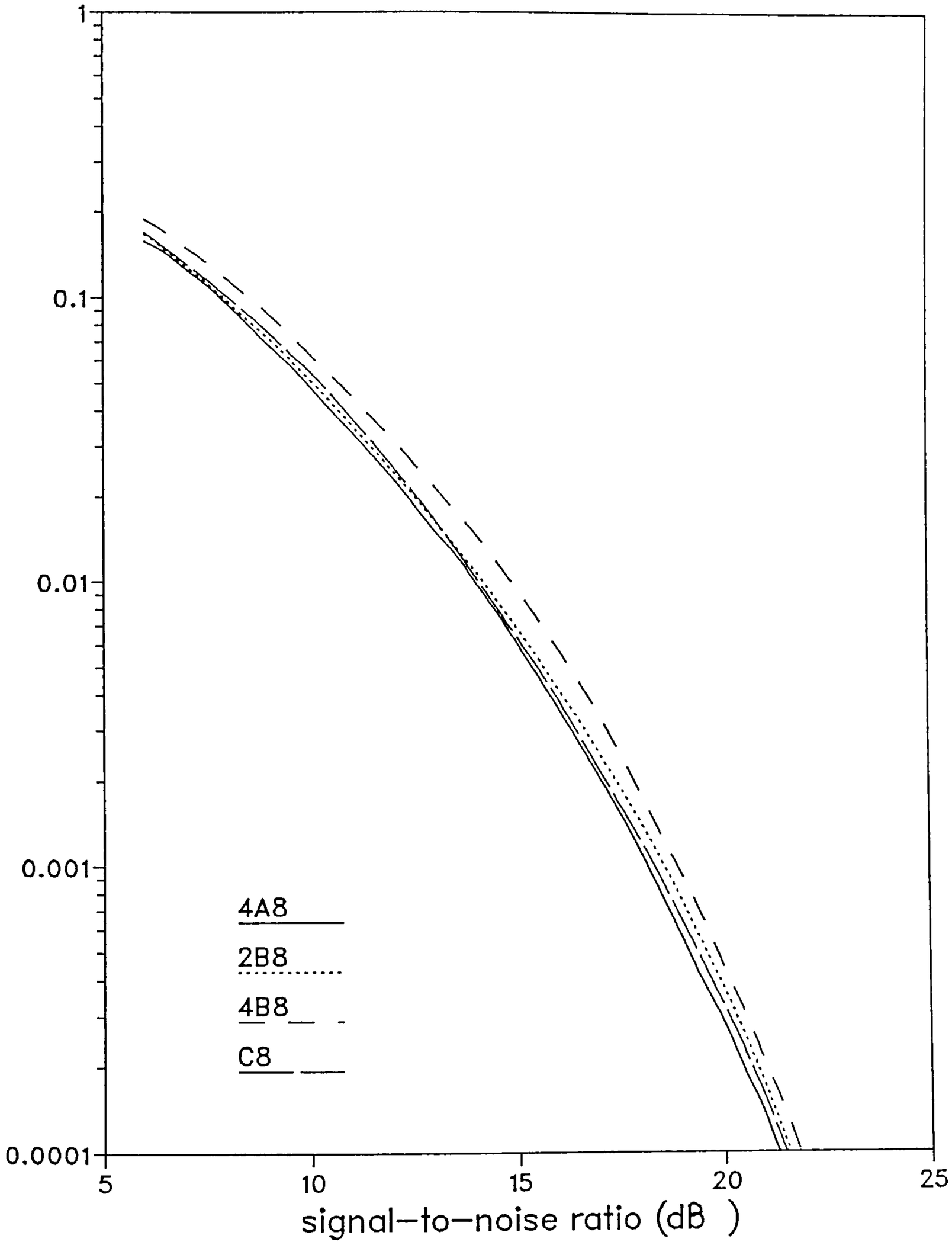


Fig.4.6.17. Detectors with 8 stored vectors operating over Channel 1

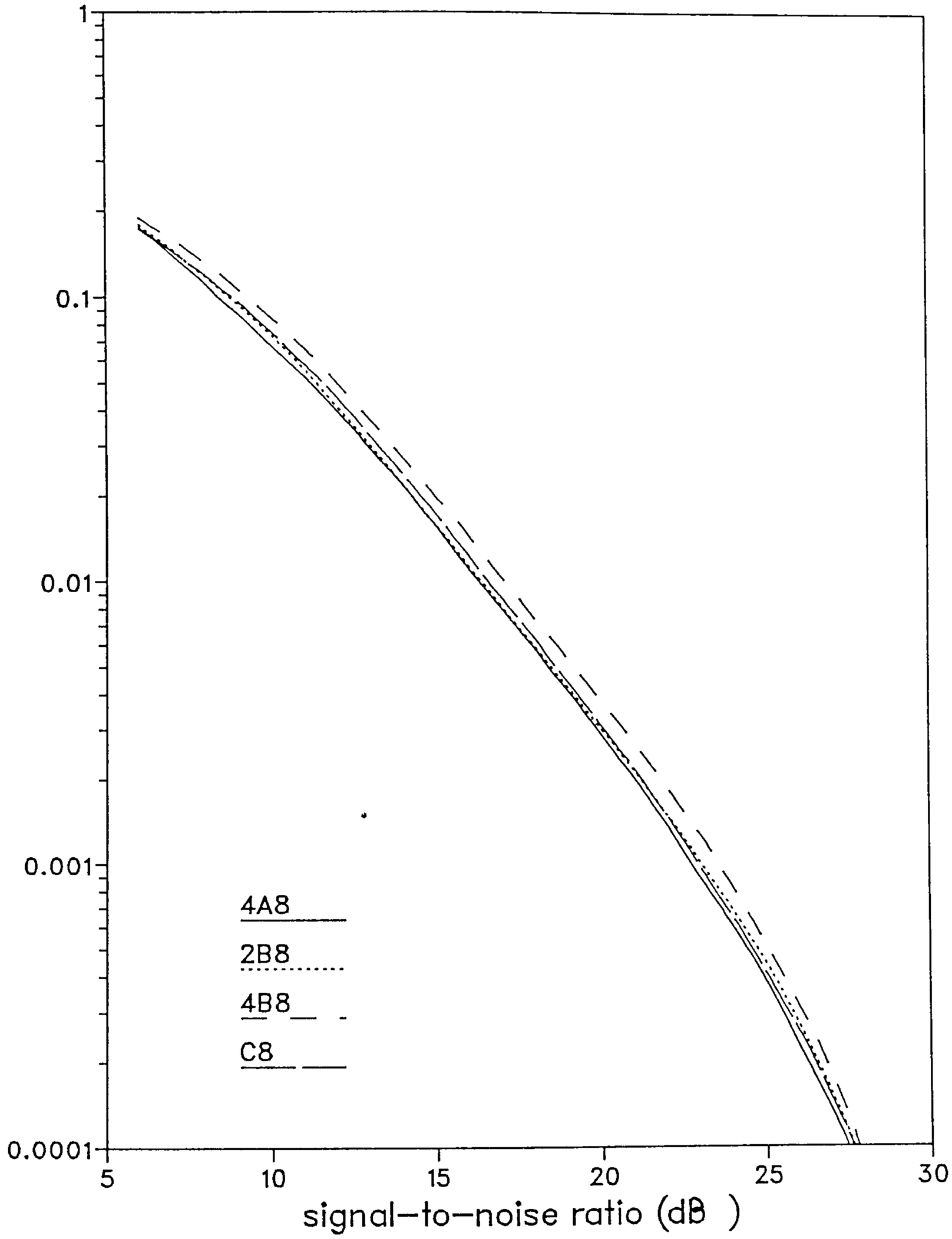


Fig.4.6.18. Detectors with 8 stored vectors operating over Channel 2

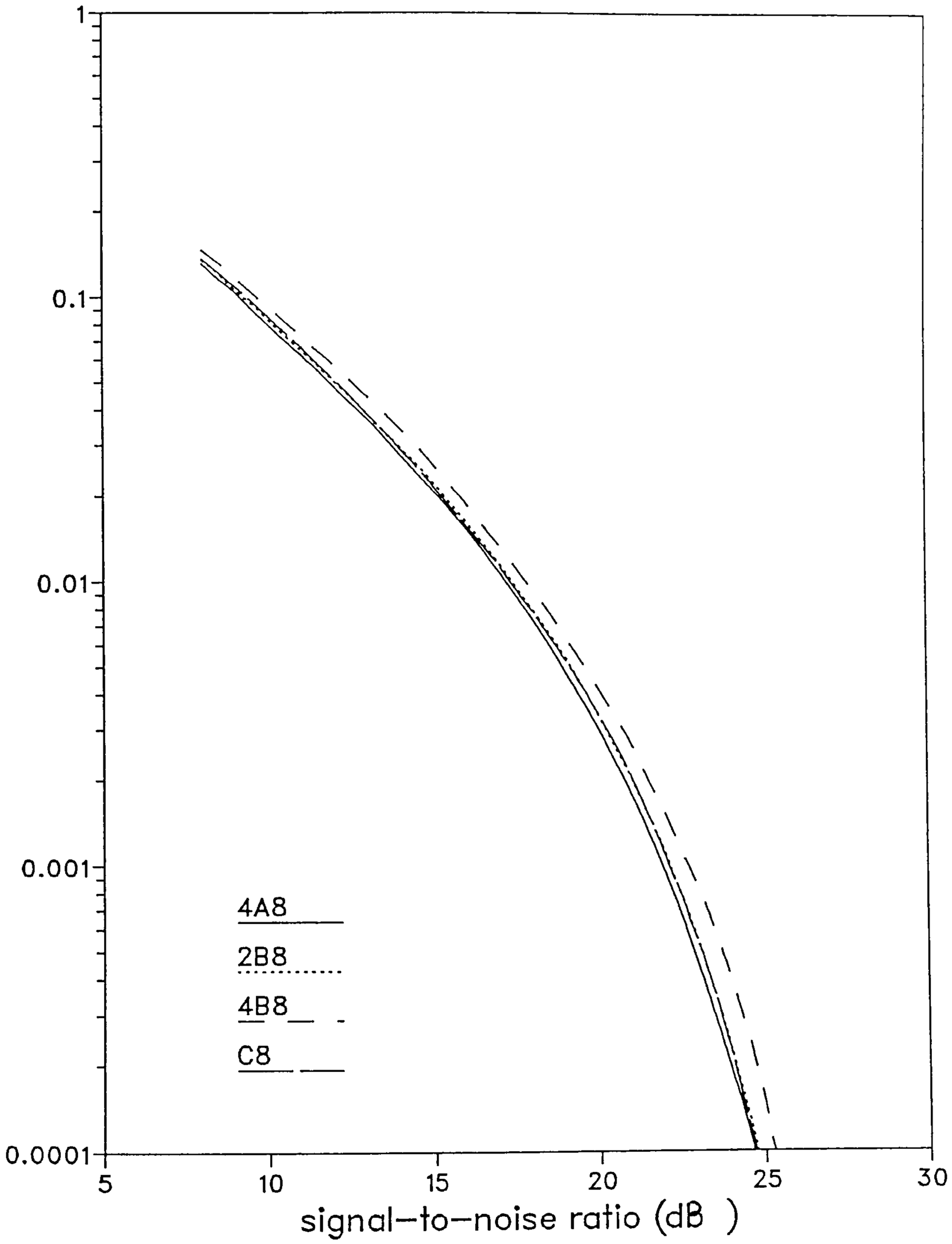


Fig.4.6.19. Detectors with 8 stored vectors operating over Channel 3

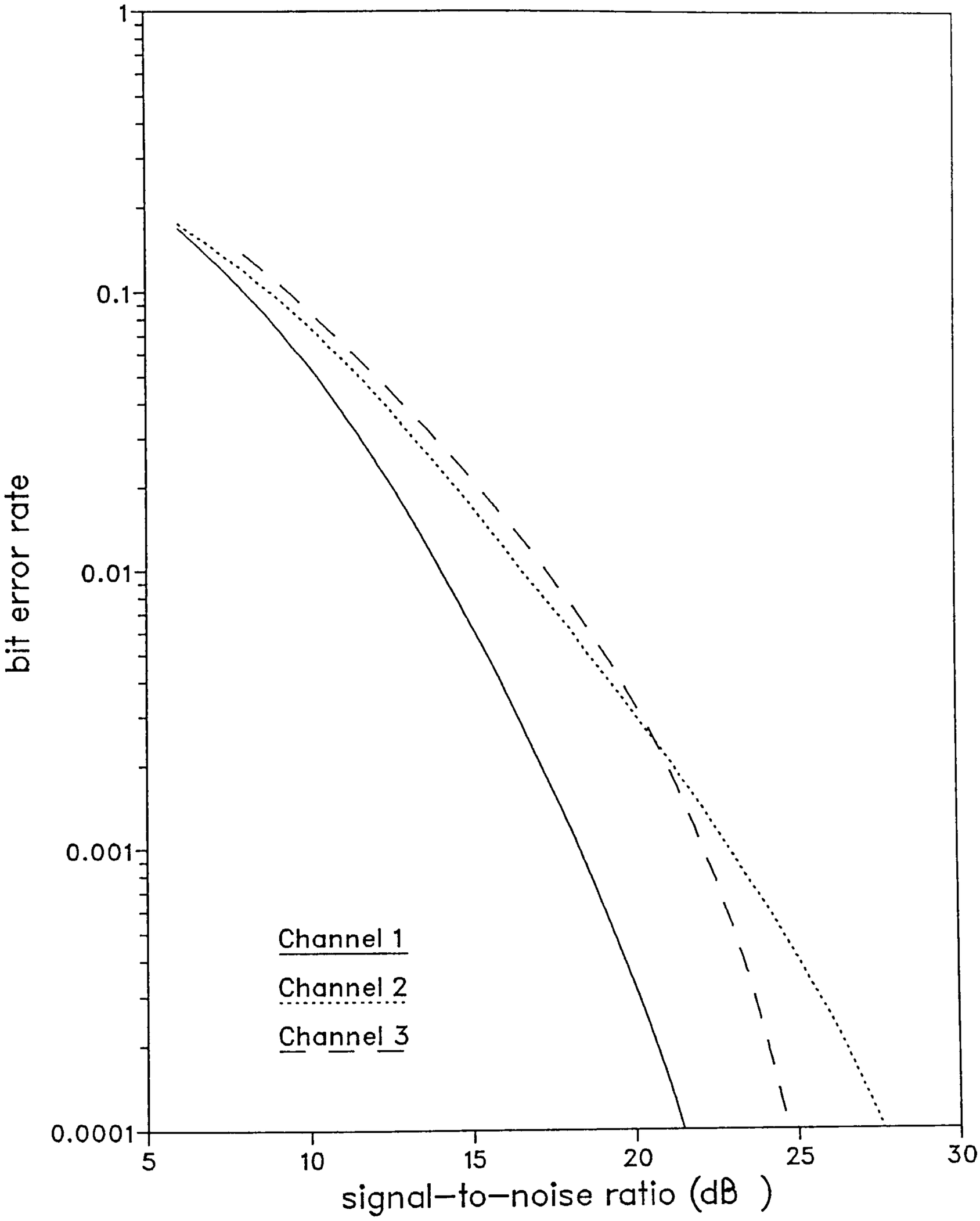


Fig.4.6.20. System C8 over the three channels

CHAPTER 5

TRELLIS CODED MODULATION FOR A 4.8 KBIT/S HF RADIO MODEM

5.1 INTRODUCTION

In the classical approach to channel coding, redundant bits are inserted in the information sequence and this redundancy is exploited at the receiver in order to detect and/or correct errors that occur due to channel noise [1-4]. The introduction of redundancy at the transmitter is carried out by a 'channel encoder' and at the receiver, a 'channel decoder' uses this redundancy to recover the received signal.

The codes that these encoders implement have been broadly categorized into two groups, namely, block codes and convolutional codes. Structures of abstract algebra are used to in the context of block codes while linear sequential circuits often lead to convolutional codes. Generally, it has been accepted that, for similar degrees of complexity of the encoder and decoder, convolutional codes outperform their counterparts, the block codes [5-7]. This is, in fact, a mistaken view [37].

Traditionally however, with both these types of code, the encoder and modulator are separated, and as such, the transmission of the redundant bits implied either a reduction in data rate given constraints on bandwidth, or, an increase in bandwidth given constraints on the data rate. Thus, for a given data rate, the performance improvement (in terms of the probability that a transmitted bit is received incorrectly, given a particular signal-to-noise ratio) that is achieved by coding is at the expense of bandwidth. More concisely, the trade off is increased power efficiency for reduced bandwidth efficiency.

At least, until 1976. In June of 1976, Ungerboeck [8] proposed the technique of combining convolutional encoding with modulation, a technique that is now widely known as trellis coded modulation (TCM) [9-10]. The codes that resulted, fall under the general category of 'trellis codes' [11-13]. The important feature of TCM however, is that the redundancy required by the coding process is obtained by the use of larger sets of channel signals than is required for uncoded transmission. Thus, the redundant bits are not transmitted in isolation, rather, they are incorporated in an expanded set of channel signals. The upshot of this is that TCM does not change the data rate and/or bandwidth relative to uncoded transmission, and it has been shown that considerable coding gains can be achieved by this method [8,10,14-22]. For bandwidth limited applications, TCM is,

indeed, an attractive proposition.

The objective of this part of the work is to examine the performance improvement, if any, that could be achieved by using TCM, for the 4.8 kbit/s HF radio modem described in Chapter 4. Since the modem is to be used for voiceband communication in the HF region, it is, necessarily, a bandwidth limited design, and since TCM does not impose the requirement of additional bandwidth, whatever coding gain that is achieved will only be at the expense of added complexity.

The chapter continues, in Section 5.2, with a detailed description of TCM, presented via an example. Section 5.3 describes the model of the data transmission system that is used in the simulation. This is equivalent to the model that was described in the previous chapter, except for the inclusion of the encoder and decoder. Section 5.4 examines in some detail, the decoders that are used in the receiver. These decoders are near maximum likelihood decoders. In Section 5.5, the results of the computer simulation tests are given and discussed.

5.2 TRELLIS CODED MODULATION

5.2.1 General Description

Trellis coded modulation (TCM) is a combination of convolutional encoding and modulation, which uses an expanded set of channel signals to transmit the information, where this expanded signal set contains the redundancy created by the process of convolutional encoding [9-11]. Thus, the convolutionally encoded signals are 'mapped' in a one-to-one fashion, onto a set of channel signals. Both the coding and mapping functions are designed with the objective of maximising the minimum Euclidean distance between coded channel signal sequences. (The Euclidean distance between any two code sequences is defined as the square root of the sums of the squares of the geometric distances between corresponding symbols of the two sequences). As a result, the minimum Euclidean distance (or, free distance) of the coded channel signal sequences is greater than that of the corresponding uncoded channel signal sequences. Since channel signals with a large Euclidean distance are more resistant to additive noise than those with a small Euclidean distance, then, all other factors being equal, a performance advantage could be gained by the use of TCM, at least over additive white Gaussian noise channels.

A generalized representation of an encoder-modulator for TCM is shown in Fig.

5.2.1 [10]. Assume that the design constraints require that m bits must be transmitted per modulation interval. In TCM, any such group of m bits will (in most cases [9]) be incorporated in one of a set of 2^{m+1} channel signals, with the transmission of one channel signal occurring per modulation interval. Let these m bits (at t=iT) be denoted as,

$$f_i = [\, f_i(m) \quad f_i(m-1) \quad ..... \quad f_i(m'+1) \quad f_i(m') \quad ..... \quad f_i(1) \,] \tag{5.2.1}$$

The trailing m' of these m bits ($m' \leq m$) are passed through a convolutional encoder which gives at its output, the sequence

$$v_i = [\, v_i(m'+1) \quad v_i(m') \quad ..... \quad v_i(1) \,] \tag{5.2.2}$$

The output of the convolutional encoder, along with the (m-m') bits which did not pass directly through the encoder, now form a sequence v_i^f, of {(m'+1) + (m-m')} bits, i.e., (m+1) bits, giving rise to a rate - m/(m+1) convolutional code where,

$$v_i^f = [\, v_i(m+1) \quad v_i(m) \quad ..... \quad v_i(m'+1) \quad v_i(m') \quad ..... \quad v_i(1) \,] \tag{5.2.3}$$

The (m+1) bits that constitute the sequence v_i^f are now passed through a 'signal mapper' which performs a one-to-one mapping function such that the bits are mapped onto one of a set of redundant 2^{m+1} channel signals. Of course, when the channel signal constellation is two dimensional, the signals themselves are complex valued.

The mapping process itself is integrated with the concept of set partitioning [10]. In set partitioning, a signal set of size 2^{m+1} is divided, successively, into smaller subsets with maximally increasing minimum Euclidean distance between the signals within a subset. As such, a signal set of 2^{m+1} channel signals would be partitioned, successively, into 2, 4, 8, ..., subsets with 2^m signals in each of the first two subsets, 2^{m-1} signals in each of the next four subsets, and so on. This partitioning is repeated (m'+1) times until there are $2^{m'+1}$ unique subsets with $2^{m-m'}$ different signals within each subset. Now, the mapping is performed such that the bits in the convolutionally coded sequence v_i, (see Fig. 5.2.1), determine one of $2^{m'+1}$ subsets, while the uncoded (m-m') bits determine one of $2^{m-m'}$ signals within a particular subset. This form of one-to-one mapping is commonly referred to as 'mapping by set partitioning'.

The technique of mapping by set partitioning is illustrated in Fig. 5.2.2 for an 8-PSK signal constellation. Of course now, $2^{m+1} = 8$ and m = 2, and the resulting TCM scheme is generally referred to as 8-PSK coded modulation. Since m = 2, the 8-PSK channel signal set is used to transmit two information bits per modulation interval.

Fig. 5.2.2 gives an example of the way in which subsets are created. At the first partitioning stage, subsets B0 and B1 are created, each having four channel signals. There are 4 subsets (C0, C1, C2 and C3) at the second partitioning stage with each subset comprising two channel signals, and at the final partitioning stage, 8 subsets (D0-D7) are created, each of which has a single channel signal. As mentioned earlier, the original signal set A0 need be partitioned only (m'+1) times and as such, the partitions D0-D7 need be created only if m'=m=2, i.e., when both information bits pass through the convolutional encoder. If, on the other hand, m'=1, then, only one information bit passes through the convolutional encoder, and the signal set A0 need be partitioned only twice, i.e., up to the creation of subjects C0-C3.

Another proviso in mapping by set partitioning is that, with every successive partition, the subsets created should consist of channel signals with increasing minimum Euclidean distance. This is confirmed with reference to Fig. 5.2.2. In the original signal set A0, the minimum Euclidean distance is, of course, the distance between any two successive signal points. Thus, if Δ_0 is the minimum Euclidean distance for the signal set AO, then,

$$\Delta_0 = 2\,E_{s,c}\,\sin(\pi/8) \approx 0.765\,E_{s,c} \qquad (5.2.4)$$

where $E_{s,c}$ is the average energy of the coded signals. Similarly, for subsets B0 and B1, the minimum Euclidean distance is

$$\Delta_1 = \sqrt{2}\,E_{s,c} \approx 1.414\,E_{s,c} \qquad (5.2.5)$$

and for subsets C0-C3, the minimum Euclidean distance is

$$\Delta_2 = 2\,E_{s,c} \qquad (5.2.6)$$

Thus it can be seen that,

$$\Delta_2 > \Delta_1 > \Delta_0 \tag{5.2.7}$$

and,

$$\Delta_{i+1} = \sqrt{2}\,\Delta_i \tag{5.2.8}$$

for $i > 0$. Eqn. 5.2.8 holds, in general, for two dimensional modulation schemes of the m-ary QASKtype [10]. It should also be mentioned here that, if uncoded 4-PSK were used to transmit two information bits per modulation interval, the minimum Euclidean distance would always be

$$\Delta_u = \sqrt{2}\,E_{s,u} \tag{5.2.9}$$

where $E_{s,u}$ is the average energy of the uncoded 4-PSK signals. If now, the average signal energies of the coded 8-PSK and uncoded 4-PSK schemes were the same (i.e. $E_{s,c} = E_{s,u}$), then from eqns. 5.2.5 and 5.2.9,

$$\Delta_1 = \Delta_u \tag{5.2.10}$$

that is, the minimum Euclidean distance of subsets B0 and B1 is exactly equal to the minimum Euclidean distance of the uncoded 4-PSK signal constellation.

The actual mapping of the binary sequence v_i^f (eqn. 5.2.3) onto the 2^{m+1} channel signals can now be carried out by any unique mapping function (straight binary, Gray coding, etc.), as long as the convolutional encoder connections conform to a given set of rules for maximizing the minimum Euclidean distance between the coded signals, for the selected mapping function. This set of rules refer to the partitioned subsets and as such, 'mapping by set partitioning' ensures good distance properties in the resulting codes.

An appropriate rate-m/(m+1) convolutional code may now be designed according to a set of rules which refer to the partitioned subsets. In the sequel, a degree of familiarity with the basic concepts and definitions of convolutional encoding [1] is assumed.

In TCM, a convolutional encoder is regarded simply as a finite state machine with a given number of states and a specified set of state transitions. These state transitions constitute a trellis diagram, giving rise to the term 'trellis' in the general description of such codes. If the code is of rate - m/(m+1), there are m input bits to, and m+1 output bits from the encoder at each modulation interval. As such, each state would give rise to 2^m possible transitions to a successor state, where these transitions correspond to the possible combinations of the m input bits. Parallel transitions (those starting and ending in the same state) are allowed and there may exist more than one transition per pair of states. Moreover, these state transition (trellis) diagrams should exhibit a reasonable degree of symmetry.

Once a suitable trellis diagram has been established, it remains to assign channel signals from the redundant 2^{m+1} signal set, to the various state transitions, such that the resulting code maximizes the minimum Euclidean distance between the coded signal sequences. This assignment of channel signals is carried out in accordance with the set of rules given below, which refer to the partitioned subsets [9].

1. Transitions originating from, or terminating in the same state, are assigned channel signals from any one of the subsets of size 2^m.

2. Any parallel transitions are assigned channel signals from any one of the subsets of size 2^{m-1}.

3. All channel signals should occur with equal frequency and the trellis diagram should exhibit a reasonably degree of symmetry.

Transitions originating from the same state imply that multiple error events could occur and parallel transitions imply that single error events could occur. Thus, the first two rules ensure good distance properties for these error events while the last rule ensures that the codes themselves exhibit regular structures.

It is helpful at this stage, to consider an example [9] of the design of a trellis code for TCM. Assume, as before, 8-PSK coded modulation and consider a convolutional encoder with 8 states and a state transition diagram as given in Fig. 5.2.3. Note first, that there are no parallel transitions in the trellis diagram, implying of course, that in the final set of subsets, there is only one channel signal in each subset. Thus, $m' = m = 2$, and the output, v_i, of the convolutional encoder uniquely determines not only the subset, but also the channel signal, and the resulting code has a rate of 2/3. The channel signals assigned to each state transition are given in Fig. 5.2.3, on the corresponding branch of

the trellis. It can be seen that this assignment of channel signals has been carried out in accordance with the three rules mentioned earlier. (Note that rule 2 is not applicable since there are no parallel transitions).

A convolutional encoder corresponding to this state transition diagram can be realized either as a feedback-free minimal encoder or a systematic encoder with feedback[9]. The former realization is depicted in Fig. 5.2.4. Assuming this realization, the state at t = iT is defined as,

$$z_i = [\, f_{i-2}(1) \quad f_{i-1}(2) \quad f_{i-1}(1) \,] \tag{5.2.11}$$

Accordingly, the states in the state transition diagram can now be labelled by a sequence of three binary digits. These are given in Fig. 5.2.3 in square brackets.

A convolutional encoder for TCM could thus be completely specified by means of a truth table giving the state transitions and the channel signals assigned to these transitions. An example of such, for the code described above, is given in Table 5.2.1 where, for convenience, decimal notation is used instead of sequences of binary numbers.

The mapping function assumed in this work first determines the value k', where,

$$\begin{aligned} k' = {} & 2^m v_i(m+1) + 2^{m-1} v_i(m) + \ldots + 2^{m'+1} v_i(m'+2) \\ & + 2^{m'} v_i(m'+1) + \ldots + 2^0 v_i(1) \end{aligned} \tag{5.2.12}$$

k' is thus, an integer in the range 0 to $(2^{m+1} - 1)$. The mapper then determines the phase angle ϕ_i, of the channel signal in the complex number plane, via,

$$\phi_i = \frac{\pi k'}{2^m} \text{ radians} \tag{5.2.13}$$

The channel signal s_i, is then defined as

$$s_i = \cos\phi_i + j \sin\phi_i \tag{5.2.14}$$

where $j = \sqrt{-1}$. In the case of the given example,

$$k' = 4\,v_i(3) + 2\,v_i(2) + v_i(1) \tag{5.2.15}$$

$$\phi_i = \frac{\pi k'}{4} \tag{5.2.16}$$

and

$$s_i = \cos\frac{\pi k'}{4} + j \sin\frac{\pi k'}{4} \tag{5.2.17}$$

As mentioned earlier, an important consideration of trellis codes is their free Euclidean distance. This is denoted d_{free}, where,

$$d_{free}^2 = \min_{s_i \neq s_i''} \left\{ \sum_{i=0}^{\infty} |s_i - s_i''|^2 \right\} \tag{5.2.18}$$

and s_i and s_i'' are two possible channel signals. The minimum in eqn. 5.2.18 is taken over all pairs of encoded sequences and $|s_i - s_i''|$ denotes the geometric distance between the two channel signals s_i and s_i''. Various methods have been proposed for the calculation of d_{free} for codes with a large number of states [9,18]. However, for codes with a small number of states (say, 8 or less), d_{free} can be calculated by referring to the state transition diagram, as follows. Take the all zero path as the reference path and determine the path which diverges from the all zero path at $t = iT$ and remerges at $t = jT$ $(j > i)$, with the least Euclidean distance. As long as the code is linear, then the Euclidean distance of the path so determined, will also be the free Euclidean distance, or the free distance, of the code.

In the example given earlier, the path giving rise to d_{free} is denoted in Fig. 5.2.3 as ABCD. The free Euclidean distance can thus be calculated as,

$$d_{free} = \left\{ d^2(0,6) + d^2(0,7) + d^2(0,6) \right\}^{1/2} \tag{5.2.19}$$

where $d^2(a,b)$ is the squared Euclidean distance between channel signals a and b. From Fig. 5.2.2, $d^2(0,6) = \Delta_1$ and $d^2(0,7) = \Delta_0$. Thus,

$$d_{free} = \left(\Delta_1^2 + \Delta_0^2 + \Delta_1^2 \right)^{1/2}$$

$$= \left(2\Delta_1^2 + \Delta_0^2 \right)^{1/2} \qquad (5.2.20)$$

Substituting from eqns. 5.2.4 and 5.2.5 gives,

$$d_{free} = (4.585\, E_{s,c}^2)^{1/2}$$

$$= 2.14\, E_{s,c} \qquad (5.2.21)$$

Another important characteristic of trellis codes is their asymptotic coding gain, which is defined as,

$$G_{c,u} = 10 \log_{10} \left[\frac{d_{free,c}^2}{E_{s,c}} \Big/ \frac{d_{free,u}^2}{E_{s,u}} \right]$$

$$= 10 \log_{10} \left[\frac{d_{free,c}^2}{d_{free,u}^2} \Big/ \frac{E_{s,c}}{E_{s,u}} \right] \qquad (5.2.22)$$

where,

$d^2_{free,c}$ = squared free Euclidean distance of the code.

$d^2_{free,u}$ = squared free Euclidean distance of the corresponding uncoded scheme for the transmission of m bits per modulation interval.

$E_{s,c}$ = average signal energy of the coded signals.

$E_{s,u}$ = average signal energy of the uncoded signals.

For m-ary PSK coded modulation schemes however, it is general practice to have $E_{s,c} = E_{s,u}$, and the work in this thesis conforms to this relationship by having,

$$E_{s,c} = E_{s,u} = \sqrt{2} \qquad (5.2.23)$$

The asymptotic coding gain is, now,

$$G_{c,u} = 10 \log_{10} \left[\frac{d^2_{free,c}}{d^2_{free,u}} \right] \qquad (5.2.24)$$

Thus, for the example given earlier, from eqns. 5.2.21 and 5.2.9,

$$G_{c,u} = 10 \log_{10} \left[\frac{4.585\, E^2_{s,c}}{2\, E^2_{s,u}} \right] \qquad (5.2.25)$$

Using the relationship of eqn. 5.2.23 in eqn. 5.2.25,

$$G_{c,u} = 10 \log_{10} \left[\frac{4.585}{2} \right]$$

$$G_{c,u} = 10 \log_{10} \left[2.2925 \right]$$

$$G_{c,u} = 3.6 \text{ dB} \qquad (5.2.26)$$

Thus, the code given in the example achieves a gain of 3.6 dB, asymptotically, over the corresponding uncoded scheme, when used over a channel which introduces additive white Gaussian noise. For the given modulation technique and given number of states, this code is optimum in the sense that it has the largest asymptotic coding gain, when used over additive white Gaussian noise channels.

5.2.2 Rotationally Invariant Codes

When a signal is transmitted over an HF radio channel, it is subjected to large and sudden phase changes called phase jumps. The prolonged error bursts that occur due to these phase jumps can be prevented by differentially encoding the signal at the transmitter and employing the corresponding differential decoding scheme at the receiver. The above is explained in detail in Section 4.4 and Appendix B. However, in the discussion in Section 5.2.1 on the design of trellis codes for TCM schemes to be employed in HF radio modems, no allowance has been made for any likely phase jumps, which are normally manifested as fixed phase rotations of the complex valued, baseband received signal. Indeed, it has been assumed throughout that the decoder is transparent to these

possible phase rotations, i.e., the encoding process is assumed to be such that a rotated version of a coded signal sequence $\{s_i\}$ (Fig. 5.2.1), results in another valid coded sequence $\{s_i''\}$, with the corresponding decoded sequence exhibiting the same rotation.

The above, in general, is an invalid assumption for most optimum trellis codes that are designed in accordance with the three rules mentioned in Section 5.2.1. For the assumption to hold, the code should satisfy the requirement that, if a fixed phase rotation is introduced into each signal s_i, of a coded signal sequence, another valid coded signal sequence is obtained. Moreover, if properly decoded, each signal of the resulting decoded sequence should exhibit the same constant phase rotation, in the same direction. These rotations could then be removed by a differential encoding/decoding system, where the differential encoder operates on the binary data sequence at the input to the convolutional encoder at the transmitter, and the differential decoder operates on the signal at the output of the convolutional decoder at the receiver.

In designing trellis codes that meet the above requirement, another rule must be followed, in addition to the three rules mentioned in Section 5.2.1. The type of codes that result are generally referred to as rotationally invariant codes [23-26]. In order that the ensuing discussion is properly understood, it is necessary to define the term 'rotation' in the context of this chapter, lest it be confused with the rotation of a vector. Let

$$\alpha_i = [\, \alpha_i(m) \quad \alpha_i(m-1) \quad ..... \quad \alpha_i(1) \,] \tag{5.2.27}$$

be the binary sequence that is input to a differential encoder, and let

$$\beta_i = [\, \beta_i(m) \quad \beta_i(m-1) \quad ..... \quad \beta_i(1) \,] \tag{5.2.28}$$

be its output binary sequence at time $t = iT$. Assume too, that both the vectors α_i and β_i represent points on a complex number plane such that all possible points represented by α_i (or β_i) form a 2^m-ary PSK signal set. In differential encoding (see Section 4.4 and Appendix B), the phase angle of the complex valued symbol represented by β_i is given by the sum of the phase angles of the complex valued symbols represented by β_{i-1} and α_i, with the absolute values of these complex valued symbols being held constant. As such, at each time interval there may be a change in the complex valued symbol represented by β_i (and hence, in β_i), from one of its possible values to another. This change is called a 'rotation' in the complex valued symbol represented by β_i (and hence, in β_i), and the

amount of rotation is given by the particular value of α_i . Since there are 2^m possible values of α_i , so there are 2^m possible rotations in the complex valued symbol represented by β_i (and hence, in β_i). These 2^m rotations in β_i result, of course, in another valid vector β_i , representing another valid complex valued symbol. Thus, from hereon, a 'rotation' will be taken as the above mentioned change in phase of the corresponding complex valued symbol.

At this stage it is appropriate to present a formal definition for rotationally invariant codes. Consider Fig. 5.2.1. Let

$$F_n = [\, f_h \quad f_{h+1} \quad ..... \quad f_n \,] \tag{5.2.29}$$

be a possible sequence of sets of m bits that are input to a convolutional encoder where f_i , (i = h, h+1, ..., n), is as given in eqn . 5.2.1. Assume too, that each vector f_i contains differentially encoded bits, i.e., similar to the vector β_i in eqn. 5.2.28. Also let

$$D_n = [\, d_h \quad d_{h+1} \quad ..... \quad d_n \,] \tag{5.2.30}$$

be a sequence of complex valued symbols such that each d_i is the symbol representing the corresponding f_i , (i = h, h+1, ... n), in the complex number plane. Each d_i could assume any one of 2^m possible complex values, where the entire set of possible complex values form a 2^m-ary PSK signal set. Again, let

$$V_n^f = [\, v_h \quad v_{h+1} \quad ..... \quad v_n \quad ..... \quad v_{n+k} \,] \tag{5.2.31}$$

be the corresponding convolutionally encoded sequence of sets of (m+1) bits, where $v_i{}^f$, (i = h, h+1, ..., n+k), is as given in eqn. 5.2.3 and k is the memory of the code. The memory is simply the number of stages of the shift register representing the code, and is measured here in symbols. Also, let

$$S_n = [\, s_h \quad s_{h+1} \quad ..... \quad s_n \quad ..... \quad s_{n+k} \,] \tag{5.2.32}$$

be the sequence of complex valued transmitted symbols (see Fig. 5.2.1) where each symbol s_i , is uniquely determined by the corresponding (m+1)-bit sequence $v_i{}^f$. Each s_i can assume any one of 2^{m+1} possible complex values.

Now let

$$F_n^{'} = [\, f_h^{'} \quad f_{h+1}^{'} \quad ..... \quad f_n^{'} \,] \tag{5.2.33}$$

be another possible sequence of sets of m bits that are input to the convolutional encoder and let

$$D_n^{'} = [\, d_h^{'} \quad d_{h+1}^{'} \quad ..... \quad d_n^{'} \,] \tag{5.2.34}$$

be the corresponding sequence of symbols representing $F_n^{'}$ in the complex number plane. The sequence $F_n^{'}$ is derived from the sequence F_n with each component f_i, (i = h, h+1, ... n), changed such that the corresponding d_i is rotated through a fixed angle of θ radians in the complex number plane. This rotation is assumed to carry each d_i (and hence, each f_i), to another one of its possible values. Thus,

$$d_i^{'} = d_i\, e^{j\theta} \tag{5.2.35}$$

and

$$\theta = \left(\frac{2L\pi}{2^m} \right) \text{ modulo-}2\pi \quad \text{radians} \tag{5.2.36}$$

where L is an integer, $j = \sqrt{-1}$ and i = h, h+1, ..., n. Also let

$$V_n^{f'} = [\, v_h^{'} \quad v_{h+1}^{'} \quad ..... \quad v_{n+k}^{'} \,] \tag{5.2.37}$$

be the convolutionally coded sequence corresponding to $F_n^{'}$, and let

$$S_n^{''} = [\, s_h^{''} \quad s_{h+1}^{''} \quad ..... \quad s_{n+k}^{''} \,] \tag{5.2.38}$$

be the corresponding complex valued transmitted symbol sequence.

The convolutional code is then said to be transparent to the given phase rotation of θ radians if each component s_i'', $(i = h, h+1, \ldots, n+k)$, is such that

$$s_i'' = s_i e^{j\theta} \tag{5.2.39}$$

If now, eqn. 5.2.39 holds true for all possible $\{F_n\}$ and $\{F_n'\}$, the convolutional code is transparent to all 2^m possible rotations in $\{d_i\}$. The given convolutional code is then said to be rotationally invariant to angles of θ radians, where θ is as given in eqn. 5.2.36. In the above situation, if the phase of the received signal is rotated by an angle of θ radians, the correct decoded sequence at the output of the decoder, in the absence of noise, must be F_n'. This fixed phase rotation of θ radians in each component of the decoded sequence can now be removed by a differential decoding operation corresponding to the differential encoding that is performed at the transmitter.

As mentioned earlier, in order to design rotationally invariant codes, the following rule [24] must be taken into account, when assigning channel signals to the state transitions in the trellis diagram. This rule should, of course, be satisfied in addition to the three rules mentioned in Section 5.2.1.

4. Let the state transition from state z_i to state z_{i+1} correspond to the input vector f_i and the output vector $v_i^{\,f}$. Also, let z_i', z_{i+1}' and f_i' be the corresponding equally rotated versions of z_i, z_{i+1}, and f_i. Then, for a state transition from state z_i' to state z_{i+1}', with the input vector f_i', the corresponding output vector must be $v_i^{\,f'}$, where $v_i^{\,f'}$ is simply $v_i^{\,f}$ rotated by the same amount and in the same direction. This relationship is known in the sequel, as the rule of equal rotation.

Rotationally invariant codes could thus be designed in accordance with these four rules. However, if a particular implementation of the encoder (such as feedback or feedforward) is desired, further rules need to be adhered to, when designing rotationally invariant convolutional codes. These rules were first formulated by L.F. Wei [23] for m-ary QAM coded modulation, and are given in their entirety in the appropriate reference. For the purpose of the work carried out in this thesis however, the four rules mentioned previously form an adequate set of constraints for the design of rotationally invariant convolutional codes.

The rule of equal rotation gives rise to codes that are rotationally invariant to the 2^m possible rotations in d_i. However, for any rate - m/(m+1) code, there are 2^{m+1} possible

rotations in s_i, and hence, the resulting codes are only transparent to half of the possible rotations in the received signal. This fact will be further discussed in the following section, where an example of a rotationally invariant code is given.

5.3 MODEL OF DATA TRANSMISSION SYSTEM USED IN THE SIMULATION

The model of the data transmission system is given in Fig. 5.3.1 and is essentially the same as that explained in Section 4.4, except for the modifications given below and the inclusion of a convolutional encoder and signal mapper at the transmitter, and the corresponding decoder at the receiver. The model assumes a synchronous, serial data transmission system with a transmission rate of 4800 bit/s. It employs 8-PSK coded modulation such that two information bits are transmitted per modulation interval, making the signal element rate 2400 baud. The redundancy of the code is contained in the additional signal levels of the 8-PSK signal set, which requires 3 bits for the representation of each signal.

The information to be transmitted is in the form of a sequence of binary digits $\{\alpha_i\}$, where these are statistically independent and equally likely to have either of the values 0 or 1. Also,

$$\alpha_i = [\, \alpha_i(2) \quad \alpha_i(1) \,] \tag{5.3.1}$$

so that the sequence $\{ \alpha_i \}$ is divided into groups of two bits.

The $\{\alpha_i\}$ are fed to a differential encoder which gives at its output the differentially encoded bit sequence $\{f_i\}$, where f_i is as given in eqn. 5.2.1 with m=2, such that,

$$f_i = [\, f_i(2) \quad f_i(1) \,] \tag{5.3.2}$$

The relationship between α_i and f_i is given in Table 5.3.1. Notice that the relationship is slightly different to that assumed in Chapter 4, (and given in Appendix B), for the systems tested previously. This is because, in order to be consistent with the straight binary

mapping used in set partitioning, Gray coding of the information bits is no longer assumed. Also, let the sequence $\{d_i\}$ be such that each d_i represents the corresponding two element vector f_i in the complex number plane, as given in Fig. 5.3.2. Thus,

$$d_i = \sqrt{2}\left\{\cos\frac{\pi k}{2} + j\sin\frac{\pi k}{2}\right\} \qquad (5.3.3)$$

where k is an integer and $j = \sqrt{-1}$. The factor $\sqrt{2}$ is included so that the average energy per uncoded bit is the same as for the system tested in Chapter 4. This allows a valid comparison between the trellis coded systems tested in this chapter, and the uncoded systems tested in Chapter 4.

The convolutional encoder uses the input sequence $\{f_i\}$, to give at its output the sequence $\{v_i{}^f\}$, where $v_i{}^f$ is as given in eqn. 5.2.3 with m=2, so that,

$$v_i^f = [\, v_i(3) \quad v_i(2) \quad v_i(1)\,] \qquad (5.3.4)$$

The convolutional code used here is a rate - 2/3 trellis code with eight states, where the code is rotationally invariant to angles that are multiples of $\pi/4$ radians. Details of this code are given later.

The output sequence from the convolutional encoder, $\{v_i{}^f\}$, is next mapped onto an 8-PSK signal set using the technique of mapping by set partitioning, as explained in Section 5.2.1 and shown in Fig. 5.2.2. Straight binary mapping is assumed such that the output from the signal mapper is the symbol s_i, where,

$$s_i = \sqrt{2}\left\{\cos\frac{\pi k'}{4} + j\sin\frac{\pi k'}{4}\right\} \qquad (5.3.5)$$

and

$$k' = 4\,v_i(3) + 2\,v_i(2) + v_i(1) \qquad (5.3.6)$$

with $j = \sqrt{-1}$. Clearly, k' is an integer in the range 0 to 7 inclusive, and so, each s_i is one of a set of 8-PSK data symbols, as shown in Fig. 5.3.3.

Each symbol s_i , could now be assumed to be transmitted in the form of an impulse $s_i \delta(t-iT)$, (see Section 4.4), where T =1/2400. These impulses are next fed to a model of an HF radio link, which has a complex valued, time varying impulse response $y_i(t)$. The impulse response $y_i(t)$ is exactly that derived in the previous chapter (eqn. 4.4.14) and as such, will not be described again, except in reiterating that it is a minimum phased impulse response. The stream of signal elements at the output of this channel model is given by $\{s_i\, y_i(t-iT)\}$. Stationary, complex valued, zero mean, coloured Gaussian noise w(t), is added to the signal at the output of the channel model, to give the complex valued received waveform r(t). The correlation of the noise has been discussed in Section 4.4 and will not be repeated here.

The received waveform r(t), is next sampled once per data symbol at the time instants {iT}. The delay in transmission is neglected such that the first non-zero sample of the received signal waveform is assumed to arrive with no delay. The sample r_i , of the received waveform at t=iT is now given by

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_{i,h} + w_i \qquad (5.3.7)$$

where

$$y_{i,h} = y_{i-h}(hT) \qquad (5.3.8)$$

and

$$w_i = w(iT) \qquad (5.3.9)$$

It is assumed that $y_{i,h} = 0$ for h<0 and h>g, and the vector

$$Y_i = [\, y_{i,0} \quad y_{i,1} \quad \cdots\cdots \quad y_{i,g} \,] \qquad (5.3.10)$$

is taken to be the 'sampled impulse response' of the channel at t=iT.

The received samples $\{r_i\}$, are next fed to a near maximum likelihood decoder which gives at its output, the decoded sequence of binary digits, $\{f'_{i-n}\}$. Clearly, the

decoder introduces a delay of nt seconds. In the particular case considered here, the decoder is assumed to have an exact knowledge of the impulse response of the channel, i.e., perfect estimation of the channel impulse response is assumed. The decoder is described in greater length in Section 5.4.

The decoded bits $\{f'_{i-n}\}$, at the output of the near maximum likelihood decoder, are finally fed to a differential decoder. The differential decoder performs an inverse operation to that of the differential encoder at the transmitter, to give at its output, the detected binary digit sequence $\{\alpha'_{i-n}\}$. The differential decoder relationship is given in Table 5.3.2.

It now remains to describe in further detail, the trellis code that is used in the transmitter to encode the sequence $\{f_i\}$. The code used here is a rate - 2/3, rotationally invariant code that is completely described by the truth table in Table 5.3.3 where, for the sake of convenience, the decimal equivalent of the binary sequence values are given. The appropriate state transition diagram is given in Fig. 5.3.4. Notice that there are parallel transitions in the state transition diagram, inferring that each of the final subsets in the associated mapping by set partitioning process contain two channel signals. The encoder itself is shown in Fig. 5.3.5. The state of the encoder is defined as

$$z_i = [\, f_{i-3}(1) \quad f_{i-2}(1) \quad f_{i-1}(1) \,] \tag{5.3.11}$$

and is given in square brackets alongside the state transition diagram in Fig. 5.3.4. Notice that the state z_i is completely described by the second component of the input vectors f_{i-3}, f_{i-2} and f_{i-1}.

This rotationally invariant code was designed by Dr. Z.C.Zhu [24], and represents a possible optimum design for the given number of states and given coded modulation scheme. The code itself has a minimum Euclidean distance squared of $4E_{s,c}$, where $E_{s,c}$ is the average energy of a coded signal. This corresponds to an asymptotic coding gain of $G_{c,u}$, where

$$G_{c,u} = 10 \log_{10} \left(\frac{4 E_{s,c}}{2 E_{s,u}} \right) \tag{5.3.12}$$

and $E_{s,u}$ is the average energy of an uncoded signal. However, from eqn. 5.2.23 and

Figs. 5.3.2 and 5.3.3, since $E_{s,c} = E_{s,u} = \sqrt{2}$,

$$G_{c,u} = 3.0 \text{ dB} \tag{5.3.13}$$

It has been mentioned by Dr. Zhu [24] that the above asymptotic coding gain cannot be improved on, by any other rate - 2/3, 8-state rotationally invariant code. As such, the possible optimum rate - 2/3, 8-state rotationally invariant convolutional code loses only 0.6 dB, in terms of its asymptotic coding gain, when compared with the corresponding optimum but not rotationally invariant code (see Section 5.2.1), which has an asymptotic coding gain of 3.6 dB.

Confirmation of the rotational invariance of the above code may be obtained as follows. Rotate all possible bit pairs, $f_i(2)$ and $f_i(1)$, at the input to the convolutional encoder, by 0, $\pi/2$, π or $3\pi/2$ radians. (The effects of these rotations are given in Table 5.3.4). For each rotation, determine the corresponding rotation in the bit sequence $\{v_i(3)\ v_i(2)\ v_i(1)\}$, with the aid of the truth table (Table 5.3.3) and Fig. 5.3.3, bearing in mind that the entries in the truth table are in decimal notation. It could be seen that, for any given state of the encoder, any rotation of the bit pair $f_i(2)$ and $f_i(1)$ is accompanied by exactly the same rotation in the bit sequence $\{v_i(3)\ v_i(2)\ v_i(1)\}$ (and hence in the signal element s_i). Thus, the code is rotationally invariant to 0, $\pi/2$, π and $3\pi/2$ radians, and differential encoding/decoding is used to remove the said rotations, as shown in Fig. 5.3.1.

Although the above code is rotationally invariant to any of the four possible rotations in d_i (Fig. 5.3.2), the coded signal s_i , is an 8-level signal (Fig. 5.3.3), and as such, it could have 8 possible rotations, where these rotations take s_i onto another one of its possible values. Thus, the said code is rotationally invariant only to half of the possible rotations in s_i , and hence only to half of the possible rotations in the received signal r_i , leaving an irreducible phase error of $\pi/4$ radians.

There are two methods for circumventing this problem. The first (and probably the more obvious!) is to design codes that are rotationally invariant to all possible rotations in the coded signal s_i , in this case, multiples of $\pi/4$ radians. However, this results in an unacceptable degradation in tolerance to noise with respect to the optimum but not rotationally invariant codes [24,25]. The other method is to incorporate a carrier phase correction system originally proposed by Ungerboeck [10] . This is the preferred method.

In the work described in this chapter, the objective is to determine the advantage (if any) that could be gained by using TCM over HF radio links, and as such, the results in this chapter are compared with the results in Chapter 4. In both cases, perfect estimation is assumed such that any phase jumps that do occur are known at the receiver. This obviates the need for a carrier phase correction system to be used in conjunction with the rotationally invariant code tested in this work. Indeed, it may be argued that perfect knowledge of the channel at the receiver obviates the need for rotationally invariant codes themselves, but of course, this would be defeating the purpose of the stated objective! Thus, the code tested here is not accompanied by a carrier phase correction system, but is rotationally invariant only to half of the possible rotations in the coded signal s_i. The performance of the above code should give adequate indication of the extent of the improvement that could be gained by the use of TCM over HF radio channels.

5.4 THE DECODERS

5.4.1 Viterbi algorithm decoders

The optimum decoding scheme for a convolutionally coded and undistorted signal, in the presence of additive white Gaussian noise, is that which minimizes the probability of decoding error [4,5,7]. When the transmitted symbols are equally likely to have any one of their possible values, the probability of decoding error is minimized when the decoded sequence is chosen such that it is the sequence that is most likely to be correct. The resulting decoder is called a maximum likelihood decoder [4,27-29].

A maximum likelihood decoder can be implemented by means of the Viterbi algorithm [5] and as such, is sometimes called the Viterbi algorithm decoder. These decoders use soft decisions [30] on the received signal, in the detection of the transmitted message. The theory of Viterbi algorithm decoders is well established and the interested reader is referred to references [4], [5] and [7] for a thorough and formal description and analysis.

A drawback of Viterbi algorithm decoders is that their complexity increases exponentially with the memory of the code. For example, a binary convolutional code with a memory of k bits would require the storage and manipulation of 2^k states, in the decoder. Moreover, the above is true only for coded and undistorted signals. When the code is used together with a distorting channel in which the sampled impulse response has (g+1) components as in eqn. 5.3.10, the corresponding Viterbi algorithm decoder would require the storage and manipulation of $2^k \times 2^{(m+1)g}$ states, where 2^{m+1} is the number of expanded

signal levels. Clearly, for most values of g, this is an unacceptable level of complexity.

In the three HF radio channel models used in this thesis, the values of (g+1) are 22, 22 and 20 (Section 4.6). In these circumstances, the design of Viterbi algorithm decoders becomes not merely impractical, but borders on the impossible. There are two methods for circumventing this problem [27,28,31,32]. The first is to equalize the coded and distorted signal by means of a linear or nonlinear (decision feedback) equalizer, prior to the decoding process. If ideally equalized, the input to the decoder would be a coded but undistorted signal, on which a Viterbi algorithm decoder may operate. The other method is to use a class of decoders which operate directly on the coded and distorted signal in a manner similar to that of a Viterbi algorithm decoder, with, however, a significant reduction in storage requirement and complexity. The decoders in this thesis are based on the latter method and are derived from the near maximum likelihood detectors that were used in Chapter 4. In the sequel, they will be referred to as near maximum likelihood decoders.

Two different near maximum likelihood decoders, namely, System D and System E, have been used in this work. These decoders operate on the received signal r_i, in the data transmission system model shown in Fig. 5.3.1 and explained in Section 5.3.

5.4.2 System D

System D is a near maximum likelihood decoder that is based on the Viterbi algorithm decoder for a coded and distorted signal [28] with, however, a considerable reduction in the number of stored vectors (states). The decoder operates on the received signal r_i, as given in eqn. 5.3.7, where eqns. (5.3.8 - 5.3.10) hold.

Just prior to the receipt of the sample r_i, the detector holds in store p different n-component vectors $\{Q_{i-1}\}$, where

$$Q_{i-1} = [\, x_{i-n} \quad x_{i-n+1} \quad \ldots\ldots \quad x_{i-1} \,] \tag{5.4.1}$$

and x_{i-h}, (h = 1, 2, ... , n), takes on a possible value of s_i, given the constraint below. It is also assumed that $n \geq g$ and that, when $i < 0$, $x_i = 0$.

The p vectors $\{Q_{i-1}\}$, are arranged into eight groups, where each group has p/8

vectors corresponding to a different one of the 8 states of the encoder at time t = (i-1)T. For example, when p=8, each group has a single stored vector, and the eight stored vectors in the eight groups correspond to the eight different states of the encoder. In this case (i.e., p=8), the resulting decoder becomes a Viterbi algorithm decoder for the coded but undistorted signal.

The correspondence of each group of p/8 vectors to a particular state of the encoder is such that all the vectors in the said group can take only permitted values of s_{i-1}, in determining x_{i-1}. These permitted values are, of course, the channel signals that are assigned to the state transitions corresponding to the particular state at t = (i-1)T, that is associated with the particular group of p/8 vectors. As such, the x_{i-h}, (h = 1, 2, ... , n), in any vector Q_{i-1}, can take on only one of a subset of 4 possible values of s_i, where these 4 possible values are either from subset B0 or from subset B1 (Fig. 5.2.2). The state corresponding to a group of vectors is identified by the storage locations of the vectors.

Each n-component vector Q_{i-1} is formed by the last n components of the (i-1)-component vector X_{i-1}, where

$$X_{i-1} = [\, x_1 \quad x_2 \quad \ldots\ldots \quad x_{i-1} \,] \tag{5.4.2}$$

It is evident from Section 5.3 that there is a unique one-to-one relationship between X_{i-1} and the corresponding sequence of binary digits $\{\alpha_j\}$, (j = 1, 2, ..., i-1), so that the correct detection of the $\{s_j\}$ results in the correct detection of the $\{\alpha_j\}$. Thus, each vector X_{i-1}, represents a possible sequence of transmitted data symbols $\{s_j\}$, (j = 1, 2, ..., i-1). Associated with each vector X_{i-1}, is its cost c_{i-1}, where,

$$c_{i-1} = \sum_{j=1}^{i-1} \left| r_j - \sum_{h=0}^{g} x_{j-h} y_{j,h} \right|^2 \tag{5.4.3}$$

and $x_i = 0$ for $i \leq 0$ and |u| is the absolute value of the quantity u.

On receipt of the sample r_i, each of the p vectors $\{Q_{i-1}\}$, is expanded into four (n+1)-component vectors $\{P_i\}$, where

$$P_i = [\, x_{i-n} \quad x_{i-n+1} \quad \ldots\ldots \quad x_{i-1} \quad x_i \,] \tag{5.4.4}$$

Thus, there are four vectors $\{P_i\}$, originating from any one Q_{i-1}. The first n components of such a group of four vectors are as in the original vector Q_{i-1}, from which they were derived, while the last component x_i, takes on its four different permitted values. These permitted values are given by the channel signals s_i, assigned to the allowable state transitions from the state corresponding to the original vector Q_{i-1} (see Table 5.3.3).

Thus, the 4p expanded vectors $\{P_i\}$, occur in eight groups, each with p/2 vectors and associated with a different state of the encoder at t = (i-1)T. The 4p expanded vectors $\{P_i\}$, also occur in eight groups, each with p/2 vectors and associated with a different state at t = iT. Associated with each of these 4p vectors are their costs c_i, where,

$$c_i = c_{i-1} + \left| r_i - \sum_{h=0}^{g} x_{i-h} y_{i,h} \right|^2 \qquad (5.4.5)$$

The decoder now selects, from the 4p expanded vectors $\{P_i\}$, the vector with the smallest cost. Let this vector be denoted by P_s. The first component x_{i-n}, of P_s is then used to determine the detected values $\alpha'_{i-n}(2)$ and $\alpha'_{i-n}(1)$. This is possible due to the unique one-to-one relationship (established in Section 5.3) that exists between a coded symbol s_i, and the corresponding transmitted bit pair $\alpha_i(2)$ and $\alpha_i(1)$. Any vector P_i, whose first component differs from that of P_s, is then discarded by assigning to it an arbitrarily high value of cost. From the remaining vectors, (including P_s), the decoder next selects for each of the eight states of the encoder at t = iT, the p/8 vectors with the smallest cost. The first component x_{i-n}, of all the p selected vectors $\{P_i\}$, is now omitted without changing the cost of the corresponding vector, to give the set of n-component vectors $\{Q_i\}$, where,

$$Q_i = [\, x_{i-n+1} \quad x_{i-n+2} \quad \ldots\ldots \quad x_i \,] \qquad (5.4.6)$$

These p vectors $\{Q_i\}$, are now stored, along with their associated costs. To prevent cost overflow, the smallest of the costs is now subtracted from each of the costs of the p vectors $\{Q_i\}$, such that the smallest cost becomes zero. The decoder is now ready for the next detection process.

5.4.3 System E

System E is a development of System A (Section 4.5.1) and operates without reference to any of the eight states of the encoder. Just prior to the receipt of the sample r_i, the decoder holds in store p vectors $\{Q_{i-1}\}$, together with their costs, exactly as in System E. However, p may now be any suitable integer, and any number of stored vectors may correspond to any number of states at $t = (i-1)T$.

On receipt of the sample r_i, each vector is expanded into 4 vectors $\{P_i\}$. The expansion is carried out as before, on the basis of permitted values of s_i. The decoder now selects the vector P_i with the smallest cost. Let this be denoted by P_s. The value of the first component of P_s, i.e., x_{i-n}, is then used to determine the detected binary digits $\alpha'_{i-n}(2)$ and $\alpha'_{i-n}(1)$, as in System D. Any vector P_i, whose first component differs from that of P_s, is now discarded by assigning to it an arbitrarily high value of cost. From the remaining vectors (including P_s), are next selected the p vectors with the smallest costs. The p selected vectors may correspond to any of the states at $t = iT$. The first component of each selected vector P_i, is now omitted without changing its cost, to give the corresponding vector Q_i. To prevent cost overflow, the smallest cost is subtracted from all the costs of the p vectors $\{Q_i\}$, such that the smallest cost becomes zero. The p vectors $\{Q_i\}$ are then stored, along with their costs, ready for the next detection process.

5.5 COMPUTER SIMULATION TESTS AND RESULTS

Computer simulation tests have been carried out to assess the performance of 8-PSK coded modulation over HF radio channels. A comparison is made of the performance of the coded systems with that of QPSK (4-QAM) uncoded modulation where, in both cases, two information bits are transmitted per modulation interval. The signalling rate is 2400 baud, giving an information rate of 4800 bit/s. The systems have been tested over Channels 1 and 2 where the former is a 3-skywave channel and the latter, a 2-skywave channel. These channels are the same as those considered in Chapter 4, and their relevant parameters are given in Table 2.5.2.

The model of the data transmission system used is that given in Section 5.3. The detectors (decoders) at the receiver are assumed to have perfect knowledge of the channel at every sampling instant, i.e., perfect estimation of the channel sampled impulse

response is assumed. This assumption is justified in Section 5.3. As in the tests for uncoded modulation described in Section 4.6, the noise components at the input to the detector are slightly correlated. Again, the reasons for, and consequences of this correlation is explained in Section 4.6.

In all tests involving error rates, the same fading sequence has been employed for a given channel. Moreover, these sequences are the same as those used in Section 4.6 for assessing the performance of uncoded 4-QAM modulation, and are shown in Figs. 4.6.1 and 4.6.2. The noise and data sequences however, have been varied at each signal-to-noise ratio, so that the performance is not influenced by the choice of a particular sequence. The number of components, (g+1), in the sampled impulse response of Channels 1 and 2 are, respectively, 22 and 22. The bit error rate in Figs. 5.5.1 - 5.5.6 is that in the $\{\alpha'_i\}$ (Fig. 5.3.1).

On average, a total number of 5×10^6 symbols have been involved in the plotting of each curve, making their 95% confidence limits better than ±0.5 dB. The signal-to-noise ratio (SNR) is exactly the same as for uncoded QPSK (4-QAM) modulation, and is given by ψ (eqn. 4.4.17) , where,

$$\psi = 10 \log_{10} \left(\frac{E_b}{\frac{1}{2} N_0} \right) \qquad (5.5.1)$$

where E_b and N_o are as defined in Section 4.4. Notice here, that E_b (the average transmitted energy per information bit) is the same for the coded and uncoded modulation schemes since the average signal energies in both cases are the same ($E_{s,c} = E_{s,u} = \sqrt{2}$), and since both cases assume the transmission of two information bits per modulation interval.

In the sequel, the curves representing the performance of uncoded systems, are taken directly from Section 4.6. The performance curves are labelled according to the detectors (decoders) that have been adopted. These detectors (decoders) are identified as follows. In the case of the uncoded systems, the near maximum likelihood detectors are denoted as given in Section 4.6, with NLE representing the nonlinear (decision feedback) equalizer. The near maximum likelihood decoders for the coded system are identified by an alphabetic character followed by a numeral, where the alphabetic character denotes the particular system under test, and the numeral denotes the number of stored vectors in the decoder. For example, System D with 16 stored vectors (i.e., two stored vectors per

encoder state) is denoted D16.

The code used is the rate-2/3, 8-state, rotationally invariant code given in Section 5.3. In the software simulation, look-up tables are used at the encoding stage in order to obtain the coded signal s_i, from the bit sequence α_i, and again at the decoding stage in order to obtain the detected bit sequence α'_i, from the detected value s'_i. All detectors (decoders) introduce a delay of 32 sampling intervals (i.e., n = 32 symbols) in the detection and/or decoding process.

Fig. 5.5.1 shows the performance of the coded and uncoded systems over an additive white Gaussian noise channel, where the sampled impulse response of the channel is 1, 0, 0 ..... Since the channel does not introduce any distortion, a Viterbi algorithm decoder for the coded system would require only eight stored vectors, i.e., one vector for each state. This, indeed, is the decoder used for the coded system (D8), while the detector 4A16 (Section 4.6) is used for the uncoded system. These detectors (decoders) are optimum for the particular cases considered in Fig. 5.5.1, and as such, the two systems lend themselves well for comparison.

It can be seen that coded system outperforms the uncoded system for error rates below 3.5×10^{-2}. Indeed, the coding gain increases with the reduction of the error rate such that, at error rates of 10^{-2}, 10^{-3} and 10^{-4}, the corresponding coding gains are, respectively, 1.0, 2.17 and 2.67 dB. Thus, most of the theoretical asymptotic coding gain of 3.0 dB is already obtained at error rates of 10^{-4}. These results agree with those given by Dr. Z. C. Zhu [24] for the same code.

Fig. 5.5.2 shows the performance of the coded and uncoded systems when tested over a channel formed by the cascade of the transmitter (TX) and receiver (RX) filters of Table 4.4.1. The corresponding sampled impulse response of the channel is given in Table 4.6.1. The above channel introduces distortion, (arising due to the non-ideal characteristics of the filters - Fig. 4.4.5), and as a consequence, intersymbol interference. However, there is no signal fading, multipath or otherwise.

The decoders tested here are D8, D16 and D32, while 4A16 is the detector used in the uncoded system. As expected, in the case of the coded system, there is an improvement in the performance of System D as the number of stored vectors is increased from 8 to 16 to 32. This improvement however, is very marginal, with a maximum of the order of 0.45 dB at error rates of 10^{-4}.

The relative performance of the coded and uncoded system is similar to that observed in Fig. 5.5.1, albeit with a very slight degradation in the performance of the coded system with respect to the uncoded system. The coding gain achieved by the best decoder, (D32), is now 0.68, 1.9 and 2.4 dB at error rates of 10^{-2}, 10^{-3} and 10^{-4}, respectively. This slight degradation is due to the distortion introduced by the channel. However, the reduction of the coding gain is insignificant, especially at low error rates. It could thus be concluded that these filters do not significantly affect the performance of the code, in terms of its coding gain, over uncoded QPSK (4-QAM) modulation.

Fig. 5.5.3 shows the performance of the systems when operating over Channel 1. Recall here, that Channel 1 is the 3-skywave channel with fades extending upto -14 dB (Fig. 4.6.1). The coded systems tested are D8, D16 and D32 and as in Fig. 5.5.2, the performance improves with the increase of the number of stored vectors in the decoder, with a maximum improvement of 0.55 dB between D8 and D32, at error rates of 10^{-4}.

A comparison of the coded and uncoded systems shows the coded systems D16 and D32 outperforming the uncoded system, when the detector used in the latter is the nonlinear equalizer (NLE). The coding gain achieved by D32, over the uncoded system with the nonlinear equalizer, is approximately 1.0 dB at error rates of 10^{-4}, while the crossover error rate for these two curves is 1.5×10^{-1}. However, since the decoders in the coded system are of the near maximum likelihood type, so their performance should be compared with uncoded systems that incorporate similar detectors. When the uncoded system is used with the near maximum likelihood detector C8 (chosen in Section 4.6 as the detector giving the best compromise between complexity and performance), its performance is generally superior to that of every coded system, upto error rates of about 1.4×10^{-4}, when the performance curve for D32 crosses with that of C8. When compared with D32 (the best of the coded systems) the performance of the uncoded system C8 is consistently better by about 0.4 dB. Although a modest advantage, it should be recalled here, that the complexity of C8 is an order of magnitude lower than that of the coded system D32. Thus, TCM does not appear to give an advantage over uncoded modulation (indeed, it is slightly inferior!) when the systems are used over Channel 1.

Figs. 5.5.4 and 5.5.5 confirm the above observation. Fig. 5.5.4 compares the performance of Systems D and E with 16 stored vectors, against the uncoded system C8, when operating over Channel 1. The performance of the coded systems D16 and E16 are very similar, inferring that the use of System E does not improve the performance of the coded systems. Indeed, they are both shown to be inferior to the uncoded system with C8. Fig. 5.5.5 compares the performance of the best coded system (D32) with the uncoded systems 4A16 and 4A4, when operating over Channel 1. Again, the overall performance

of the uncoded systems is better than that of the coded system D32. Even the simplest of near maximum likelihood detectors in the uncoded system, 4A4, is better than the coded system D32, upto error rates of 2.5×10^{-4}.

In Fig. 5.5.6, the relative performance of the coded and uncoded systems is compared, when the systems are used over the 2-skywave channel, Channel 2. Recall here, that Channel 2 exhibits worse fades (extending upto -20dB - Fig. 4.6.2) than Channel 1, and consequently would introduce more severe distortion into the transmitted signal. The performance of the coded systems over Channel 2, relative to that of the uncoded systems, is similar (if not worse!) to their relative performances over Channel 1. The uncoded system C8, is better than the coded systems D8 and E16, upto error rates of 10^{-4}. At 10^{-4}, there is a crossover of the curves representing C8 and E16. Even the nonlinear equalizer, NLE, (operating in the uncoded system), gives a better performance than coded D8, upto error rates of 10^{-4}. The nonlinear equalizer is better than coded E16 upto error rates of 10^{-3}. These results suggest that, even over Channel 2, QPSK (4-QAM) uncoded modulation is better than 8-PSK trellis coded modulation.

The above observations should not be misconstrued as inferring the wholesale condemnation of TCM for HF radio channels. This is because, firstly, the channels used in this thesis are worst-case channels for the given channel parameters and as such, the codes are being tested under the severest of conditions. Secondly, the code that is tested is an 8-state code offering only 3.0 dB theoretical asymptotic coding gain over the corresponding uncoded system, in additive white Gaussian noise channels. This translates into a coding gain of only 2.67 dB (Fig. 5.5.1) at error rates of 10^{-4}, which is further reduced to 2.4 dB by the distortion introduced by the filters (Fig. 5.5.2). As such, a more powerful code (say, with 16 or 32 states) with a greater asymptotic coding gain, may offer an improvement in performance relative to the corresponding uncoded system, especially when tested over more typical HF channels with less severe fades. Thirdly, interleaving could be used as a means of nullifying the effects of error bursts [33,34] (common on HF channels), in conjuction with TCM, to improve the performance of the coded systems. Indeed, the absence of interleaving is probably a major cause for the poor relative performance of the coded systems.

However, it does appear that TCM loses its reputed advantage over the corresponding uncoded modulation systems, when used over HF channels with deep selective fades. A possible reason for this is that although the code is near optimum for an additive white Gaussian noise channel, it may not be so for a fading channel. Indeed, it has been suggested in a recently published paper [35], that the criterion for the design of optimum m-ary PSK trellis codes for additive white Gaussian noise channels (i.e.,

maximizing the minimum Euclidean distance, d_{free}) is not necessarily the criterion that should be considered in the design of such codes for fading channels. A companion paper[36] describes a modified method of set partitioning appropriate for the design of codes for fading channels. Although being beyond the scope of this thesis, it would be an interesting excercise to determine the performance of TCM schemes designed according to the criteria set out in the said references [35,36], over an HF channel model similar to that used here. As regards the work in this thesis however, the conclusion is that, for Channels 1 and 2, QPSK (4-QAM) uncoded modulation is the preferred system, when compared with 8-PSK trellis coded modulation.

References - Chapter 5

1. **Lin, S. and Costello, D.J.,** 'Error control coding: Fundamentals and applications', Prentice Hall, 1983.

2. **Clark, G.C. Jr. and Cain, J.B.,** 'Error correction coding for digital communications', Plenum, 1981.

3. **Michelson, A.M. and Levesque, A.H.,** 'Error control techniques for digital communications', Wiley, 1985.

4. **Proakis, J.G.,** 'Digital communications', McGraw Hill, 1983.

5. **Viterbi, A.J.,** 'Error bounds for convolutional codes and an asymptotically optimum decoding algorithm', IEEE Trans. on Information Theory, vol. IT-13, pp. 260-269, April 1967.

6. **Forney, G.D. Jr,** 'Coding and its applications in space communications', IEEE Spectrum, vol. 7, pp. 47-58, June 1970.

7. **Viterbi, A.J.,** 'Convolutional codes and their performance in communication systems', IEEE Trans. on Communications Technology, vol. COM-19, No.5, pp. 751-772, October 1971.

8. **Ungerboeck, G.,** 'Channel coding with multilevel/phase signals', IEEE Int. Symposium on Information Theory, Ronneby, Sweden, June 1976.

9. **Ungerboeck, G.,** 'Channel coding with multilevel/phase signals', IEEE Trans. on Information Theory, vol . IT-28, pp. 55-67, January 1982.

10. **Ungerboeck, G.,** 'Trellis coded modulation with redundant signal sets; Part I: Introduction, Part II: State of the art, IEEE Communications magazine, vol. 25, pp. 5-21, February 1987.

11. **Forney G.D. Jr., Gallagher, G.R., Lang G.R., Longstaff, F.M. and Quershi, S.H.U.,** 'Efficient modulation for bandlimited channels', IEEE Journal on selected areas in Communications, vol. SAC-2, pp. 632-647, 1984.

12. **Calderbank, A.R. and Mayo, J.E,** 'A new description of trellis codes', IEEE Trans. on Information Theory,' vol. IT-30, no.6, pp. 784-791, November 1984.

13. **Calderbank, A.R. and Sloane, N.J.A.,** 'New trellis codes based on lattices and cosets', IEEE Tran. on Information Theory, vol. IT-33, no. 2, pp. 177-195, March 1987.

14. **Lebowitz, S.H. and Rhodes, S.A.,** 'Performance of coded 8-PSK signalling for sattelite communications', IEEE Int. Conf. on Communications (ICC '81) record, pp. 47.4.1-47.4.8, Denver, Colerado, USA, June 1981.

15. **Hui, J. and Fang, R.J.F.,** 'Convolutional code and signal waveform design for bandlimited sattelite channels', ibid, pp. 47.5.1-47.5.10.

16. **Wilson, S.G., Schottler, P.J. and Sleeper, H.A.,** 'Rate-3/4 16-PSK phase codes', IEEE Int. Conf. on Communications (ICC '82) record, pp. 6F1.1-6F1.5, Philadelphia, PA, June 1982.

17. **IBM Europe,** 'Trellis code modulation scheme with 8-state systematic encoder and 90° symmetry for use in data modems transmitting 3-7 bits per modulation interval', Contribution D180 to the CCITT study group XVII, October 1983.

18. **Biglieri, E.,** 'High level modulation and coding for nonlinear sattelite channels', IEEE Trans. on Communications, vol. COM-32, no. 5, pp. 616-626, May 1984.

19. **Wilson, S G., Sleeper, H.A., Schottler, P.J and Lyons, M.T.,** 'Rate-3/4 convolutional coding of 16-PSK: Code design and performance study, IEEE Trans. on Communications, vol. COM-32, no.12, pp. 1308-1314, December 1984.

20. **Rhodes, S.A and Chang, P.Y.,** 'Coded M-ary modulation for FDMA sattelite communications', IEEE Int. Conf. on Communications (ICC '84) record, pp. 22.1.1 - 22.1.9, June 1984.

21. **Brownlie, J.D. and Cusack, E.L.,** 'Duplex transmission at 4800 and 9600 bit/s on the general switched telephone network and the use of channel coding with a partitioned signal constellation', British Telecom Research Labs report, 1984.

22. **Divsalar, D and Simon, M.K.,** 'Trellis coded modulation for 4800 - 9600 bit/s transmission over a fading mobile sattelite channel', IEEE Journal on selected areas in Communications, vol. SAC-5, no.2, pp.162-175, February 1987.

23. **Wei, L.F.,** 'Rotationally invariant convolutional channel coding with expanded signal space - Part I and II, IEE Journal on selected areas in Communications, vol. SAC-2, no.5, pp. 659 - 686, Sept ember 1984.

24. **Zhu, Z.C.,'** Signal design for satellite links', Ph.D. thesis, Loughborough University of Technology, 1986.

25. **Oerder, M.,** 'Rotationally invariant trellis codes for MPSK modulation, IEEE Int. Conf. on Communications (ICC '85) record, vol. 2, pp. 552 - 556, Chicago, Illinois, USA, June 1985.

26. **Zhu, Z.C and Clark, A.P.,** 'Rotationally invariant coded PSK signals, IEE Proc. Part F, vol.134, no.1, pp. 43 - 52, February 1987.

27. **Zhu, Z.C. and Clark, A.P.,** 'Near maximum likelihood decoding of convolutionally coded PSK signals', IEE Proc. Part F, vol.135, no.1, pp. 33-42, February 1988.

28. **Clark, A.P. and Kadhim, A.K.A.R.,** 'Detection of coded and distorted QAM signals', Journal of IERE, vol .58, no.4, pp. 187 - 196, June 1988.

29. **Lee, L.H.C and Farrell, P.G.,** 'Error performance of maximum likelihood trellis decoding of (n,n-1) convolutional codes: a simulation study, IEE Proc. Part F, vol. 134, no.7, pp. 673 - 680, December 1987.

30. **Farrell, P.G. and Goodman, R.F.,** 'Soft decision error control for HF data transmission', IEE Proc. Part F, vol. 127, no.5, pp. 389 - 400, October 1980.

31. **Driscoll, J.P. and Karia, N.,** 'Detection processes for V32 modems using trellis coding', IEE Proc. Part F, vol. 135, no.2, pp. 143-154, April 1988.

32. **Wesolowski, K.,** 'Performance of the receivers of trellis coded data signals

transmitted through HF channels', Proceedings of the 4th Int. Conf. on HF radio systems and techniques, pp. 305-309, London, April 1988.

33. **Mui, S.Y.,** 'The performance of pseudorandom interleaving with Viterbi decoding of convolutional codes', IEEE Int. Conf. on communications (ICC '83) record, pp. 509-513, vol.1, Boston, MA, USA, June 1983.

34. **Divsalar, D. and Simon, K.,** 'Trellis coded modulation for 4800-9600 bit/s transmission over a fading mobile satellite channel', IEEE Journal on selected areas in Communications, vol. SAC-5, no.2, pp. 162-175, February 1987.

35. **Divsalar, D and Simon, M.K.,** 'The design of trellis coded MPSK for fading channels: Performance Criteria', IEEE Trans. on Communications, vol. 36, no.9, pp. 1004-1012, September 1988.

36. **Divsalar, D and Simon, M.K.,** 'The design of trellis coded MPSK for fading channels: Set partitioning for optimum code design', IEEE Trans. on Communications, vol. 36, no.9, pp. 1013-1021, September 1988.

37. Farrell, P.G. and Clark, A.P., 'Modulation and coding - A review', International Journal of Satellite Communications', vol. 2, pp. 287-304, 1984.

		Next State z_{i+1}				Output v_i			
	Input f_i	0	1	2	3	0	1	2	3
Current State z_i	0	0	1	2	3	0	4	2	6
	1	4	5	6	7	1	5	3	7
	2	0	1	2	3	4	0	6	2
	3	4	5	6	7	5	1	7	3
	4	0	1	2	3	2	6	0	4
	5	4	5	6	7	3	7	1	5
	6	0	1	2	3	6	2	4	0
	7	4	5	6	7	7	3	5	1

Table 5.2.1 **Truth table for the 8-state, rate-2/3 convolutional code in Figs. 5.2.3 and 5.2.4**

f_{i-1}		α_i		f_i	
$f_{i-1}(2)$	$f_{i-1}(1)$	$\alpha_i(2)$	$\alpha_i(1)$	$f_i(2)$	$f_i(1)$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	0	0
1	1	1	0	0	1
1	1	1	1	1	0

Table 5.3.1 **Differential encoding of binary digits in Fig. 5.3.1**

f'_{i-1}		f'_i		α'_i	
$f'_{i-1}(2)$	$f'_{i-1}(1)$	$f'_i(2)$	$f'_i(1)$	$\alpha'_i(2)$	$\alpha'_i(1)$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	0	0

Table 5.3.2 **Differential decoding of binary digits in Fig. 5.3.1**

		Next State z_{i+1}				Output v_i^f			
Input	f_i	0	1	2	3	0	1	2	3
Current State z_i	0	0	1	0	1	0	2	4	6
	1	2	3	2	3	1	3	5	7
	2	4	5	4	5	3	1	7	5
	3	6	7	6	7	2	0	6	4
	4	0	1	0	1	6	4	2	0
	5	2	3	2	3	7	5	3	1
	6	4	5	4	5	1	3	5	7
	7	6	7	6	7	0	2	4	6

Table 5.3.3 **Truth table for 8-state, rate-2/3, rotationally invariant code in Fig. 5.3.4**

f_i		Phase Rotation							
$f_i(2)$	$f_i(1)$	0		$\pi/2$		π		$3\pi/2$	
0	0	0	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	0	0
1	0	1	0	1	1	0	0	0	1
1	1	1	1	0	0	0	1	1	0

Table 5.3.4 **Effect of different phase rotations on f_i**

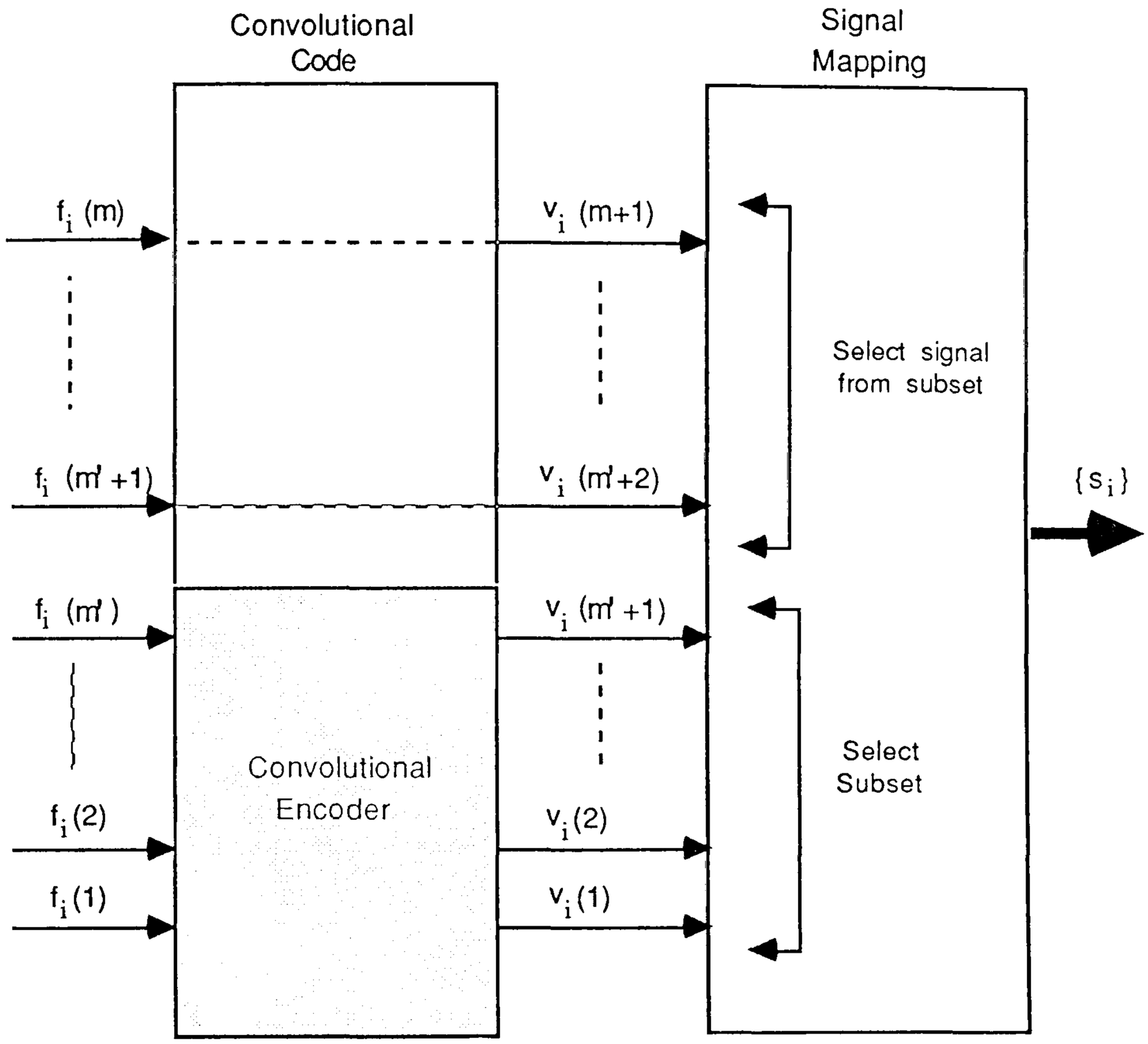
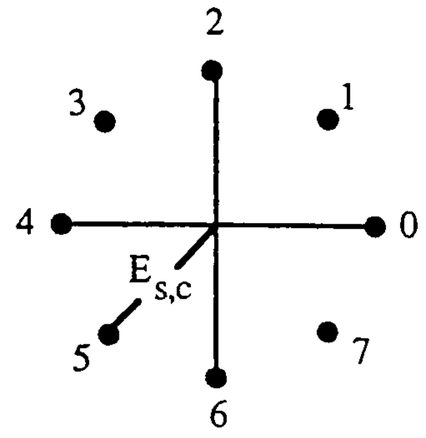


Fig. 5.2.1 **Generalized representation of Trellis Coded Modulation**



$\Delta_0 = 2E_{s,c} \sin(\pi/8)$

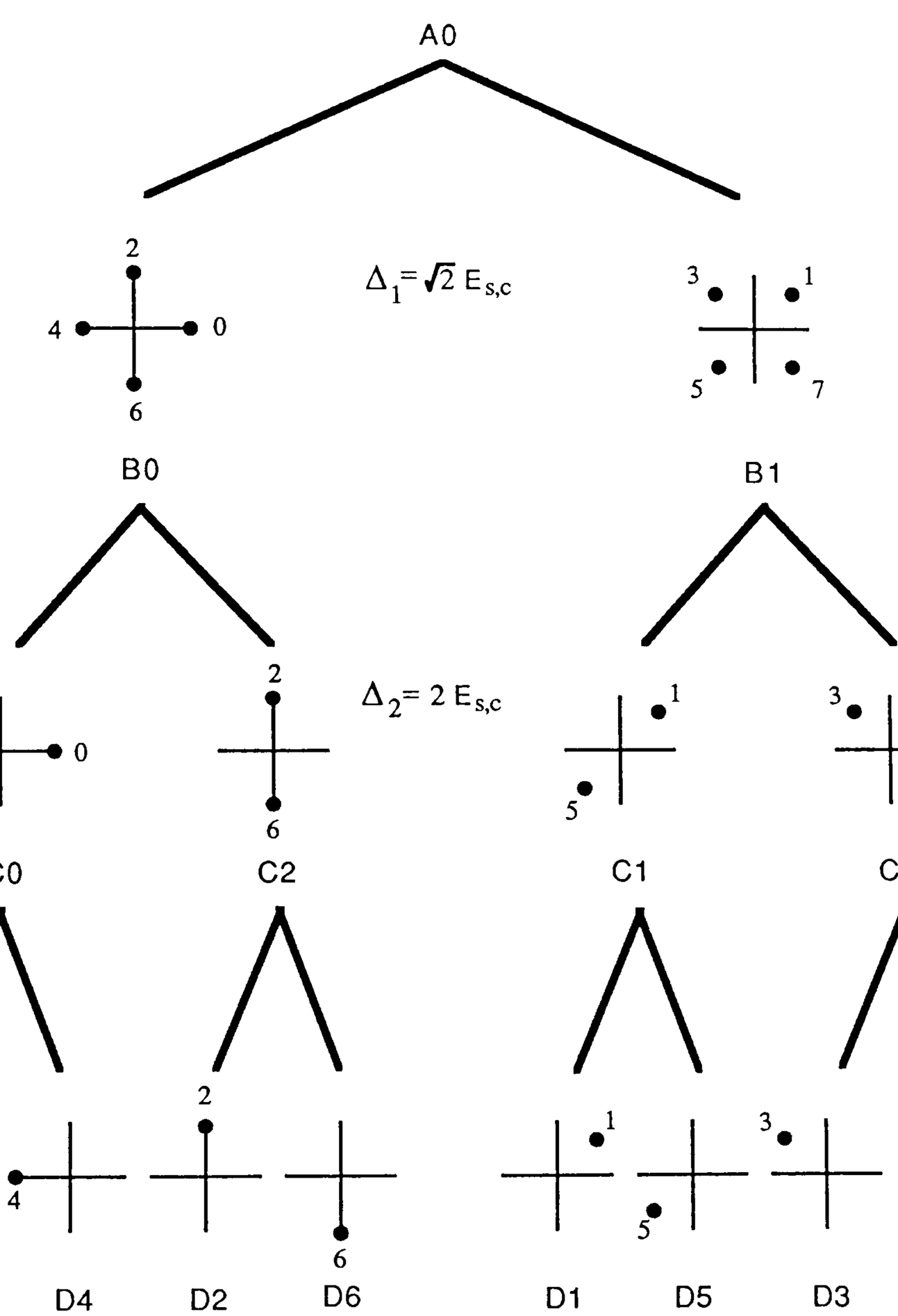


Fig. 5.2.2 **Partitioning of an 8 - PSK signal constellation**

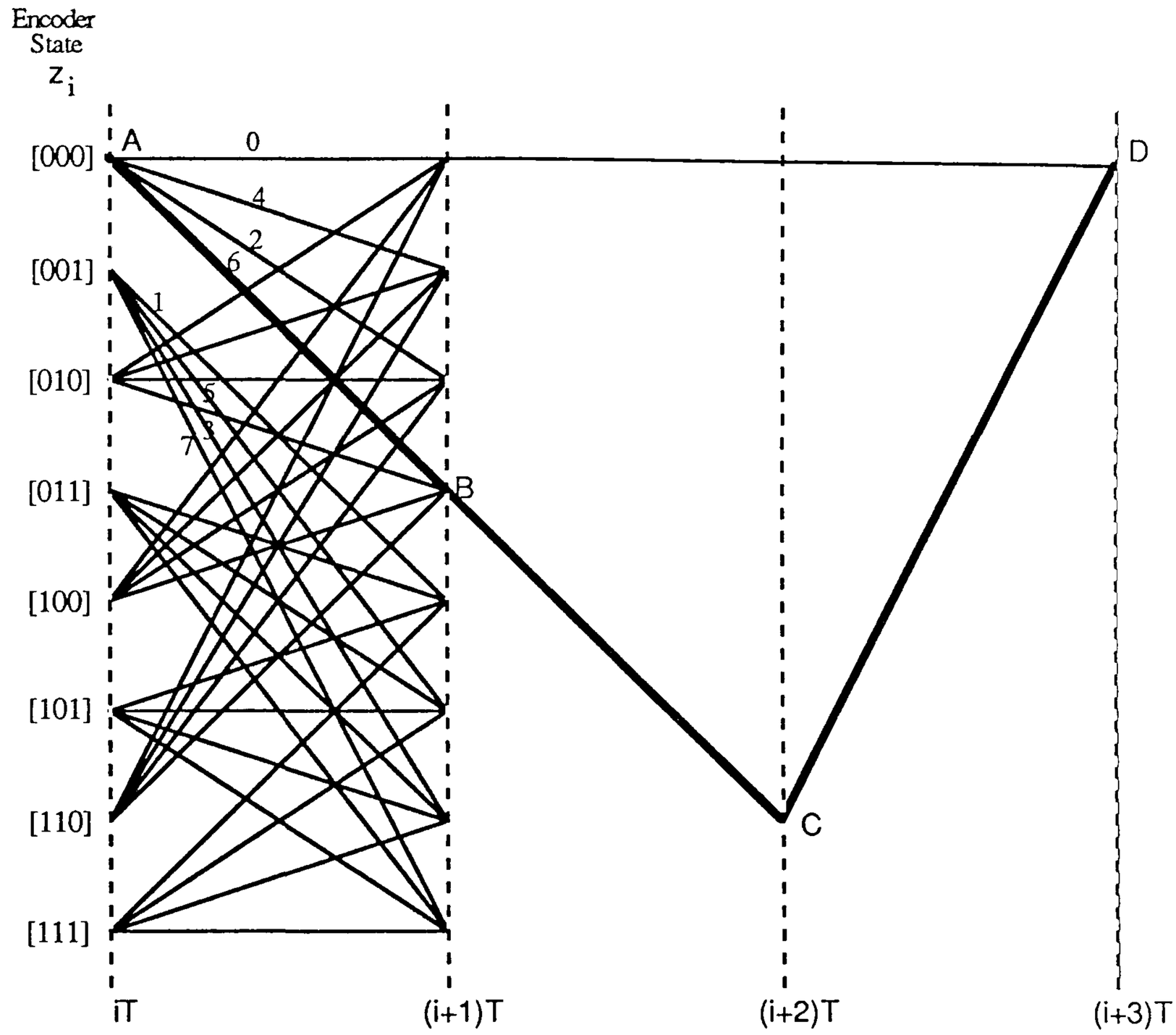


Fig. 5.2.3 State transition diagram for an 8-state, rate-2/3 convolutional code for coded 8PSK modulation

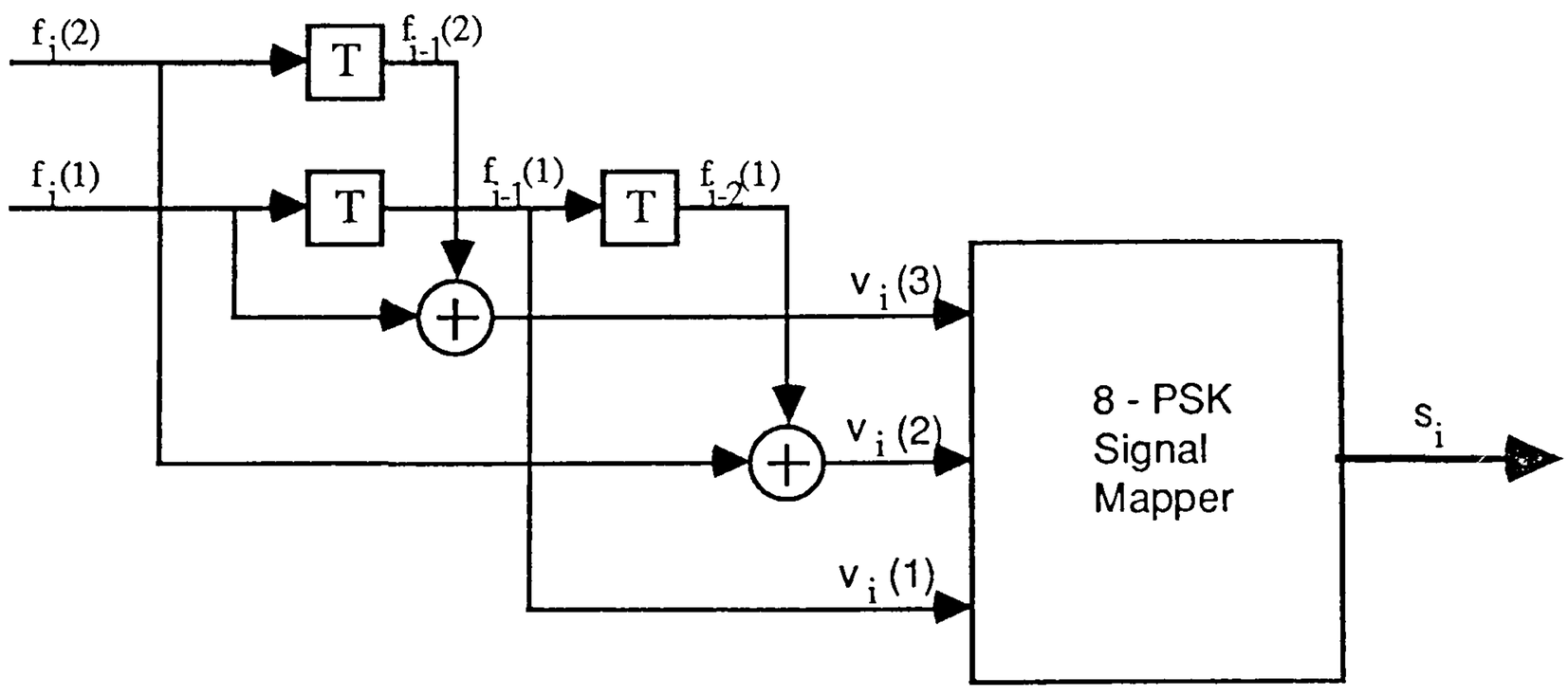


Fig. 5.2.4 Realization of trellis code in Fig. 5.2.3 as feedback free minimal encoder

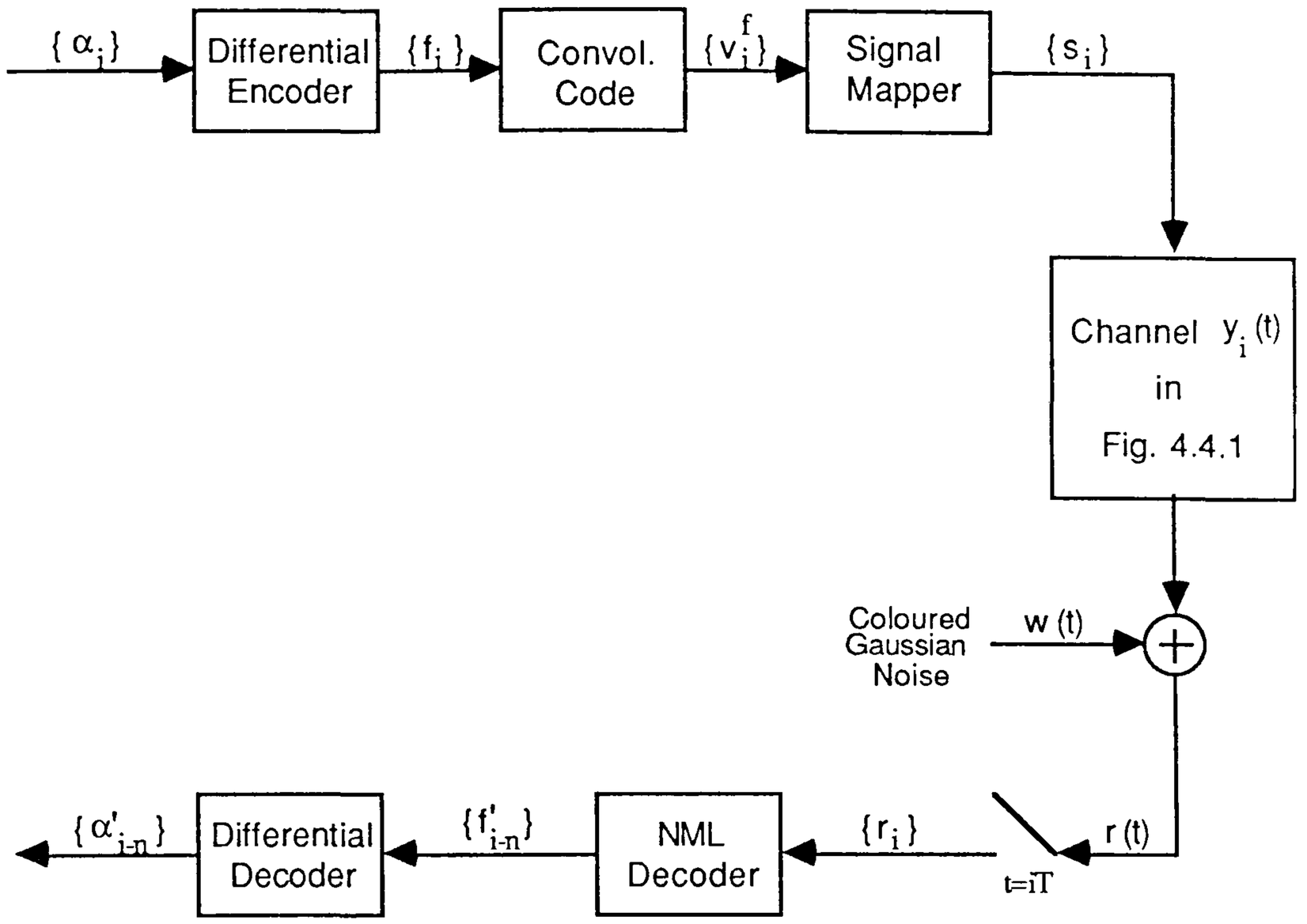


Fig. 5.3.1 **Model of the data transmission system used in the simulation tests**

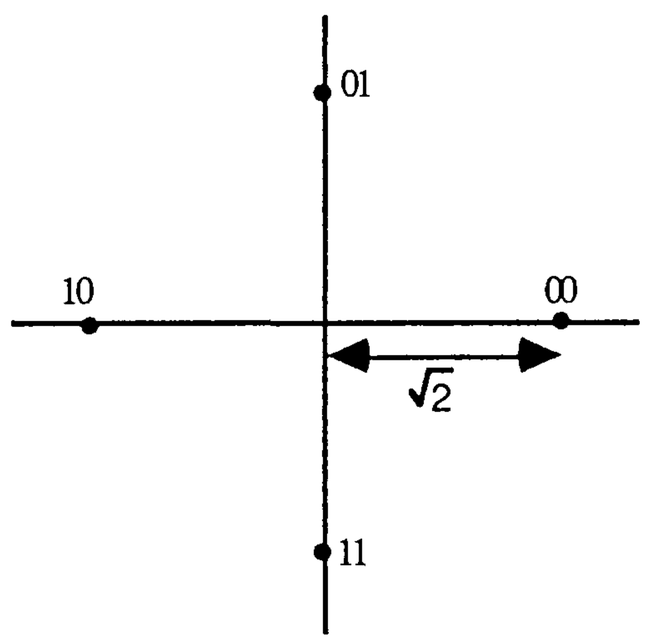


Fig. 5.3.2 f_i **represented in the complex number plane as** d_i

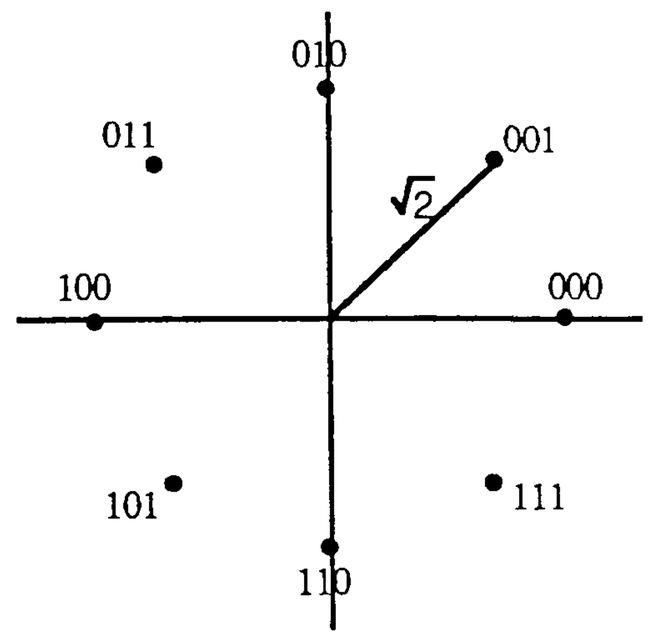


Fig. 5.3.3 **Coded signal mapped onto complex number plane as** s_i

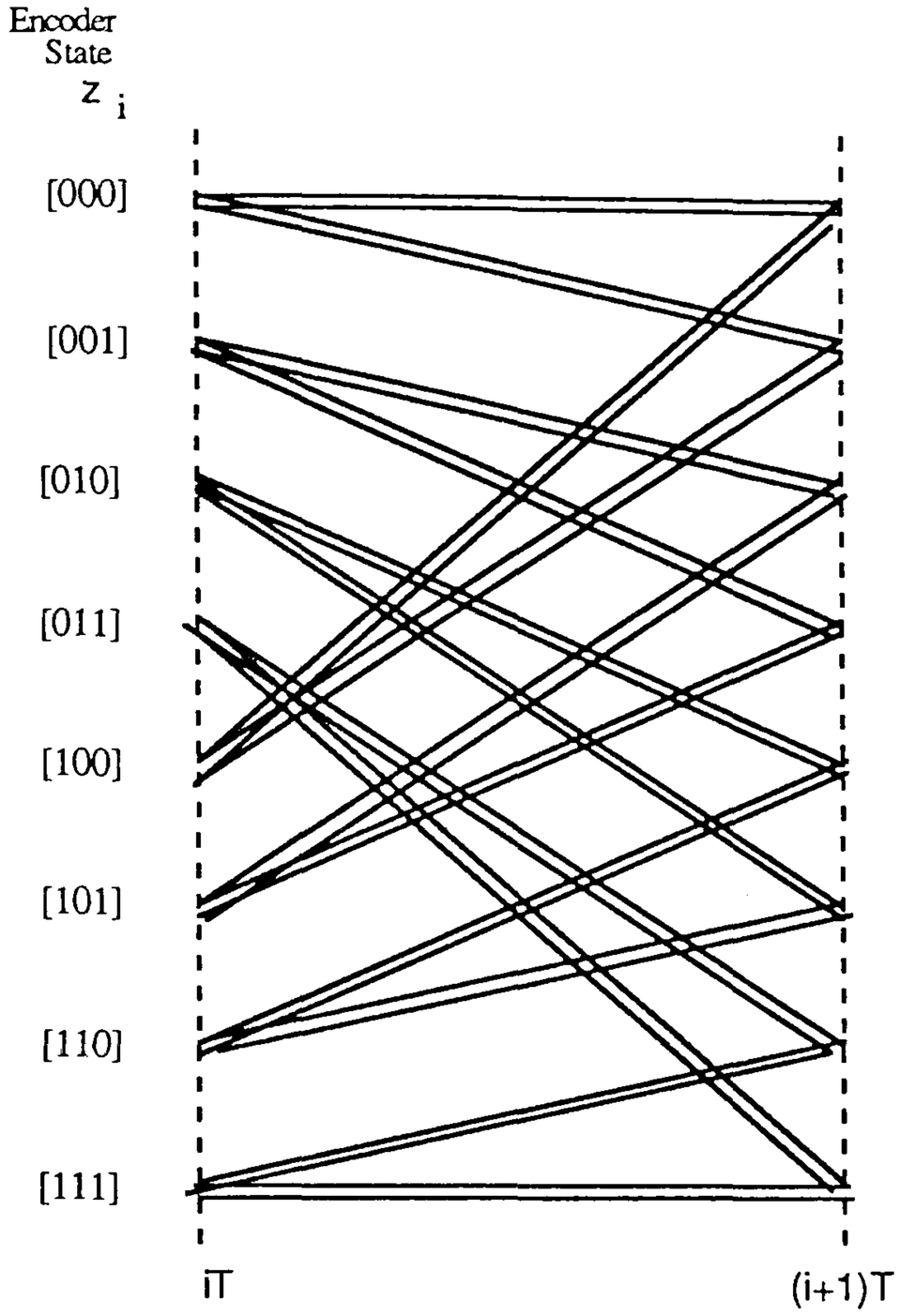


Fig. 5.3.4 State transition diagram for an 8-state, rate-2/3 rotationally invariant convolutional code

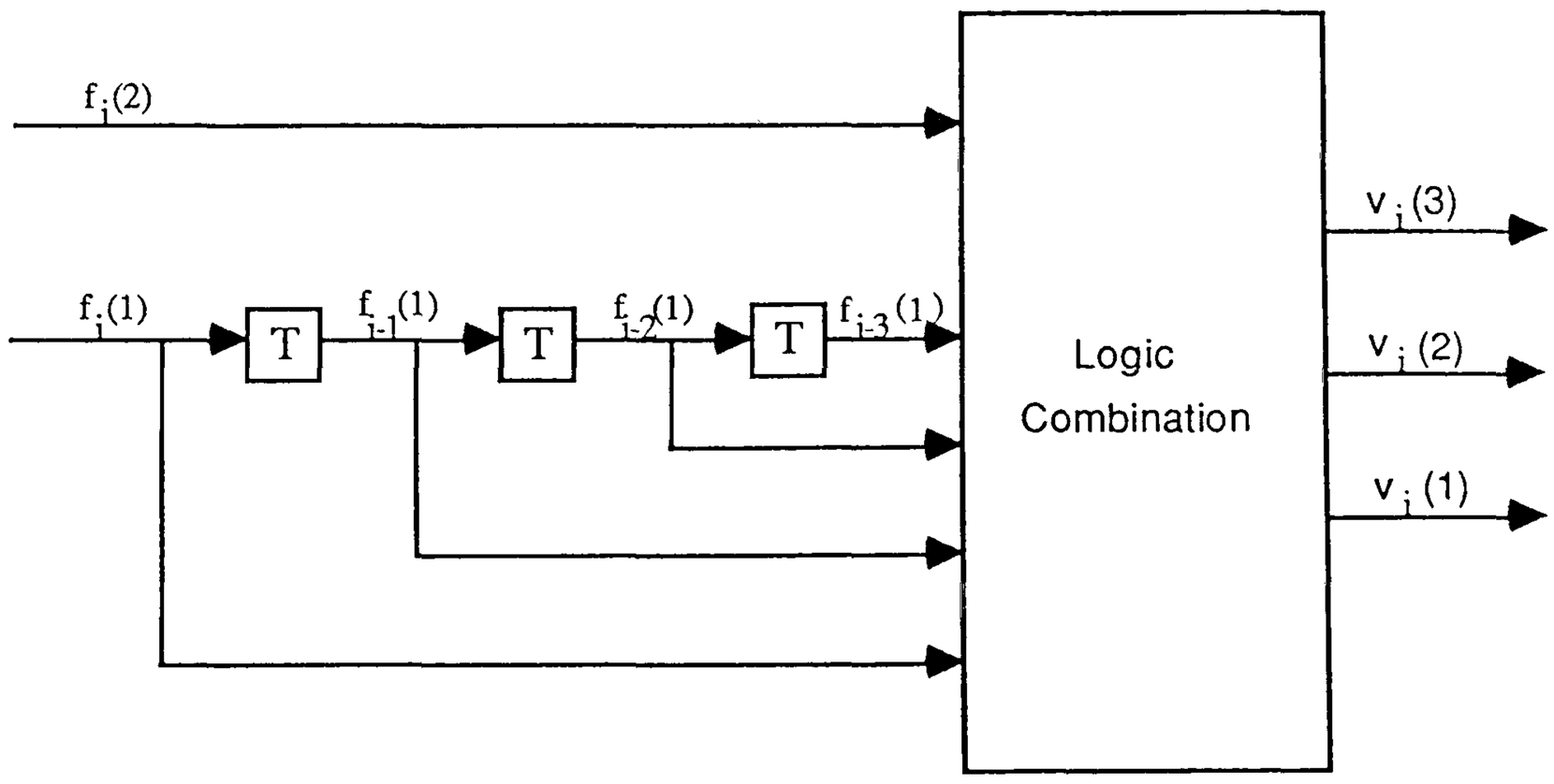


Fig. 5.3.5 An 8-state, rate-2/3, rotationally invariant coder

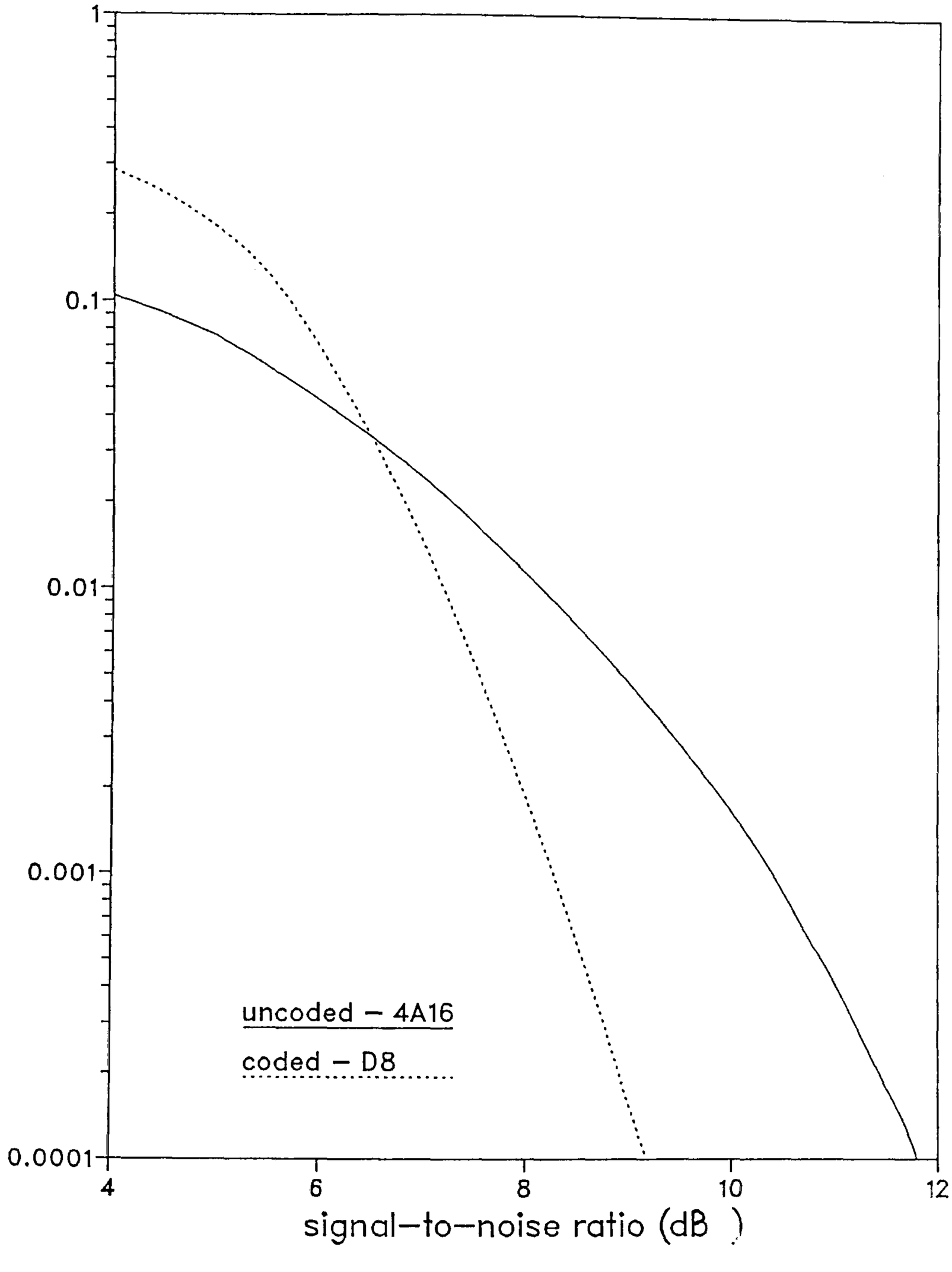


Fig.5.5.1. Performance comparison of coded and uncoded systems over additive white Gaussian noise channel

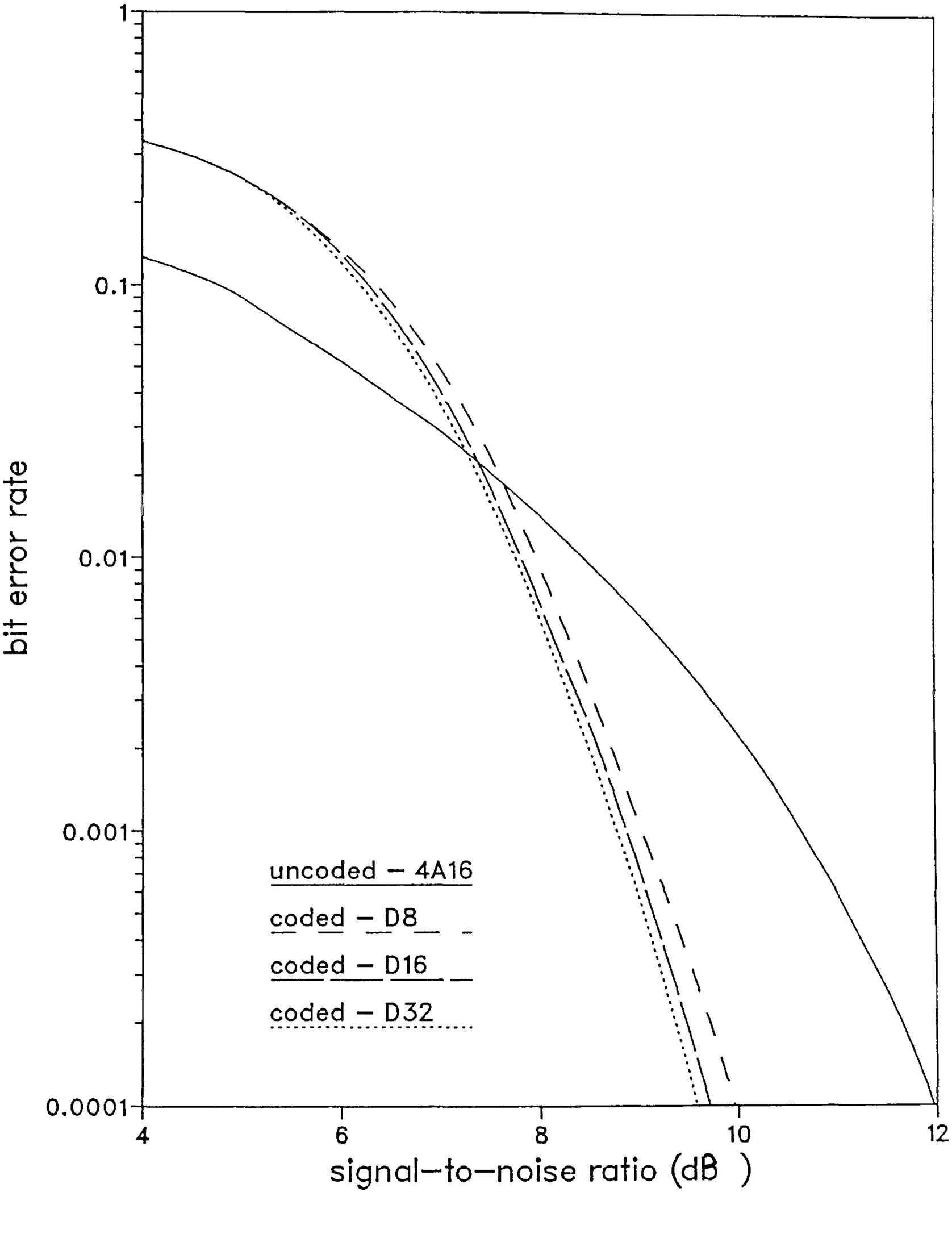


Fig.5.5.2. Performance comparison of coded and uncoded systems over channel with TX and RX filters back–to–back

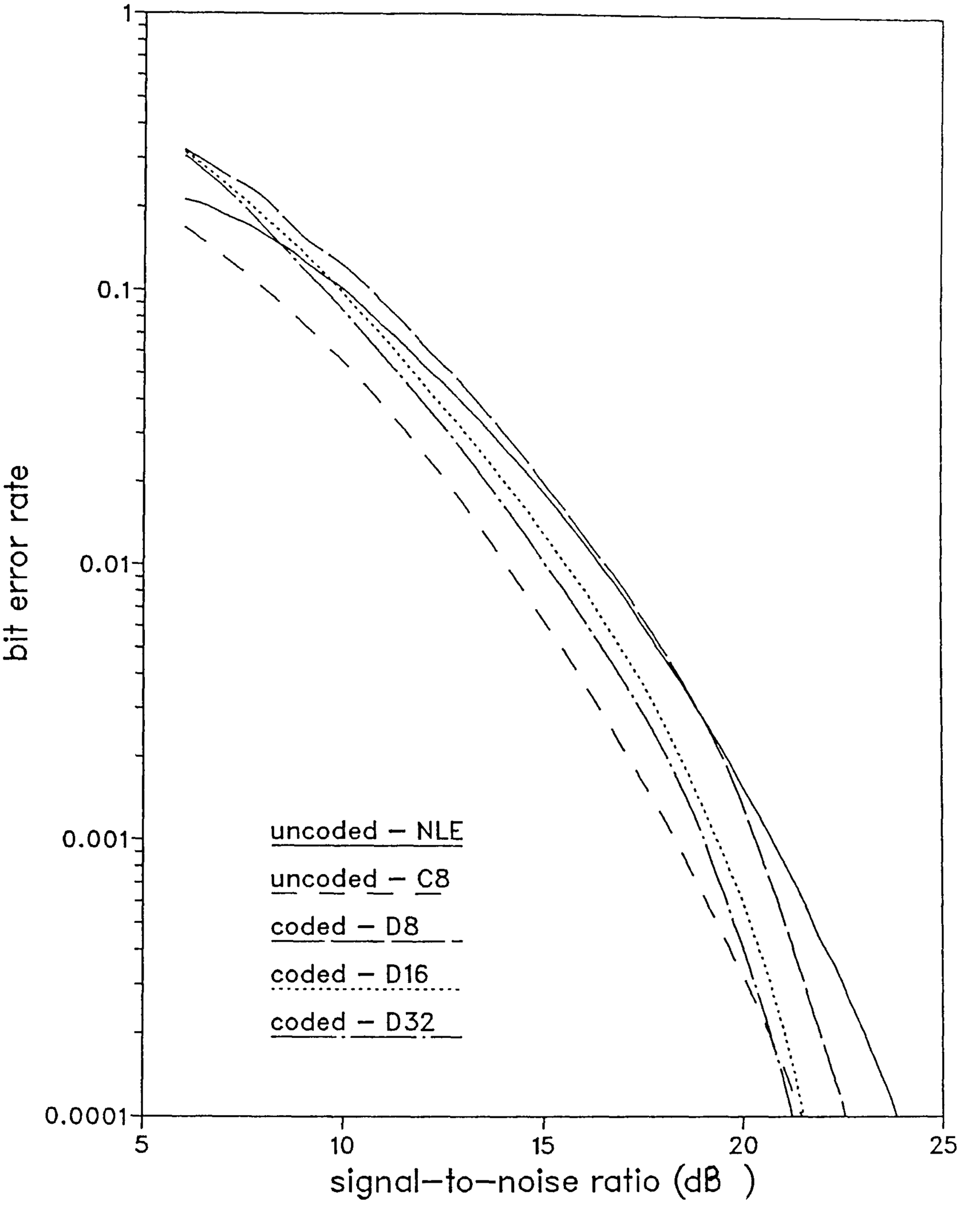


Fig.5.5.3. Performance of System D and comparison with uncoded system over Channel 1

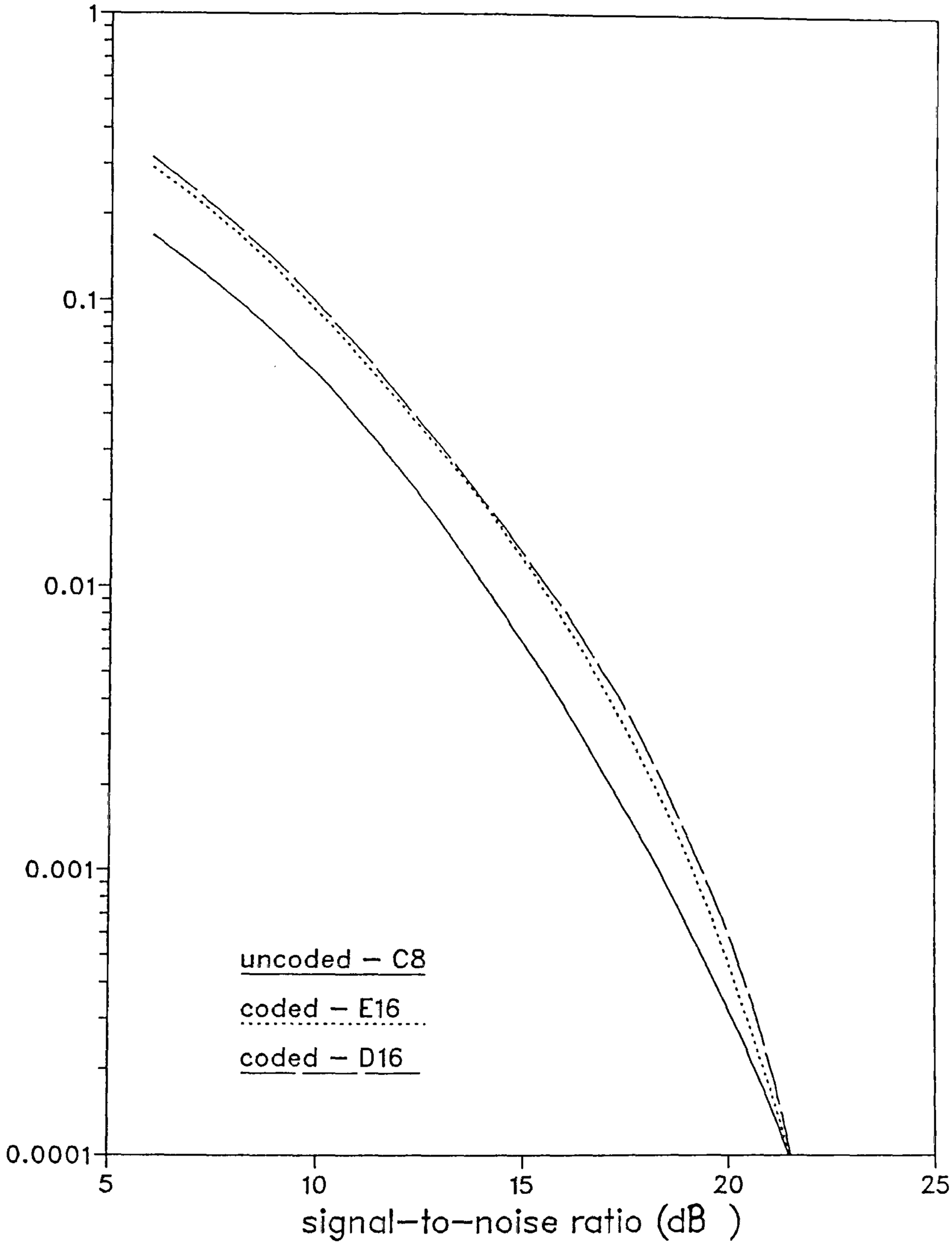


Fig.5.5.4. Performance of System E16 and comparison with System D16 over Channel 1

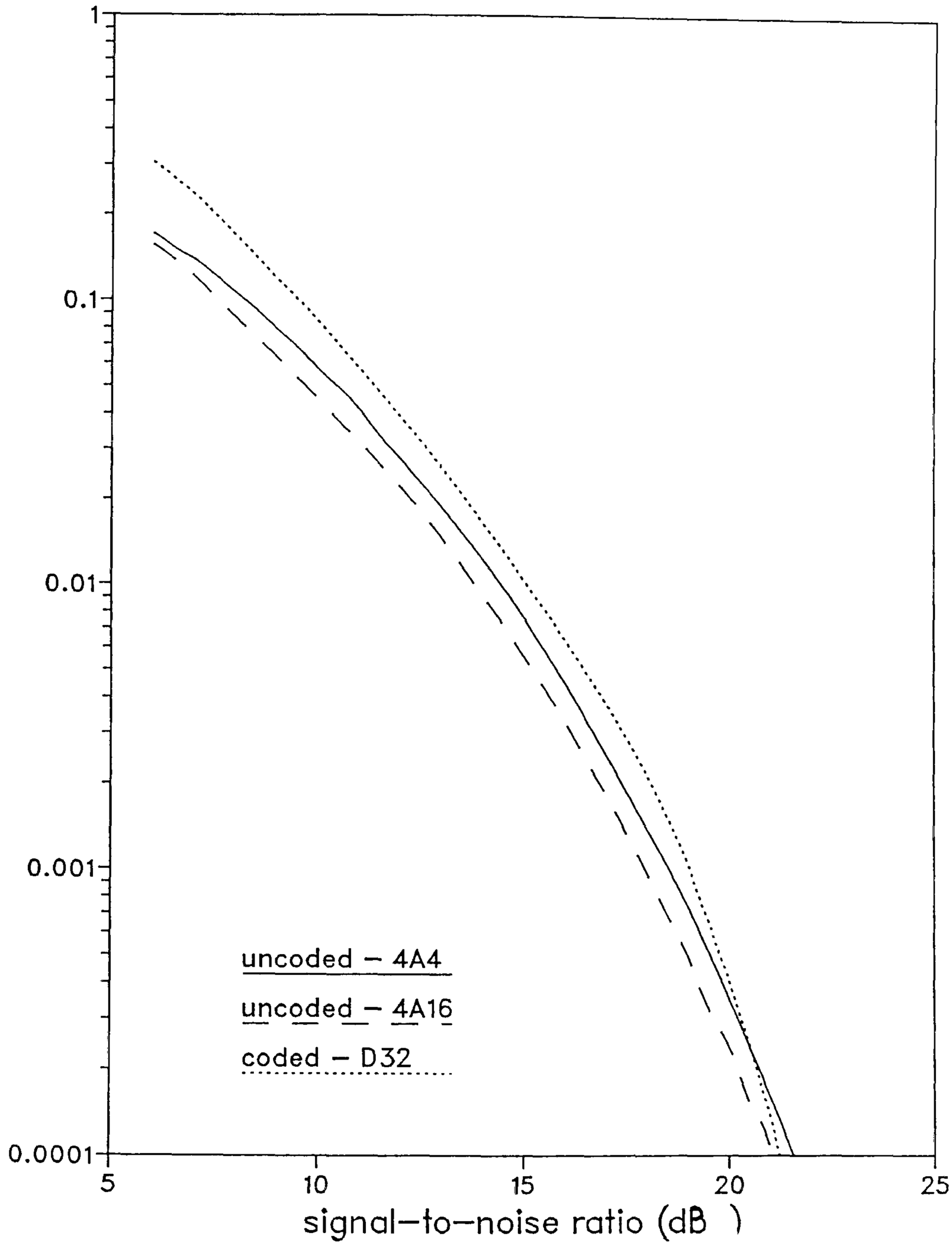


Fig.5.5.5. Performance comparison of coded System D32 with uncoded System A over Channel 1

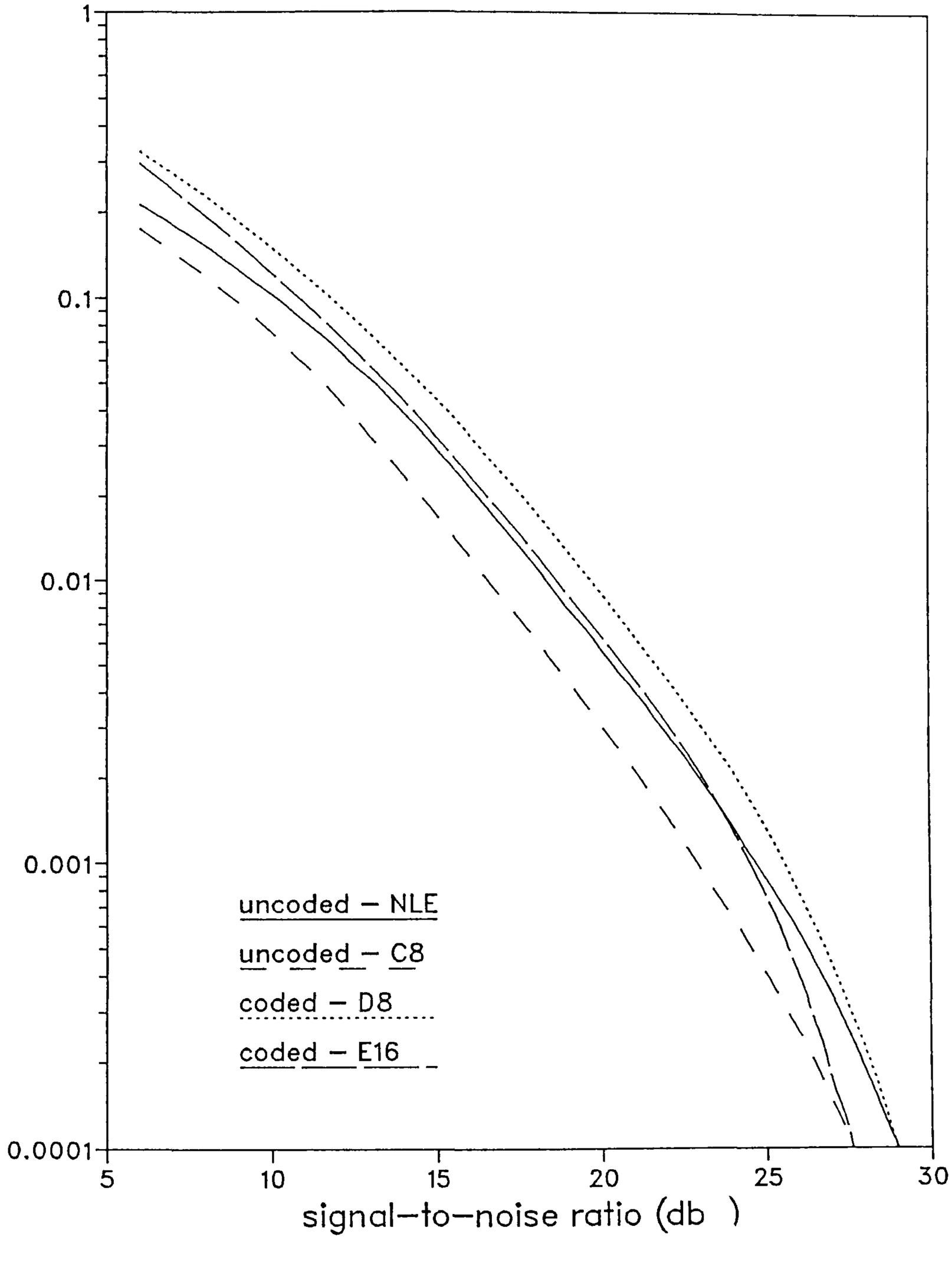


Fig.5.5.6. Performance of coded and uncoded systems over Channel 2

CHAPTER 6

CHANNEL ESTIMATION FOR FAST FADING CHANNELS

6.1 INTRODUCTION

The near maximum likelihood (NML) detectors used in the previous chapters are assumed to have an exact knowledge of the time variations introduced by the radio channel. This means that perfect estimation of the sampled impulse response (SIR) of the channel is assumed. In practice however, the detector would have to be given the sampled impulse response of the channel at every sampling instant. The device which does this is known as a channel estimator.

Channel estimators are implemented by means of adaptive filters, where the adaptive filters are used in the context of system identification, rather than their more common application of adaptive equalization [1-4]. Many such channel estimators have been designed and their performance over HF radio channels is well documented in the published literature [5-11]. However, under conditions of very fast fading, such as would occur in an urban land mobile radio environment where transmission frequencies are in the region of 900 MHz. [12], the performance of a channel estimator may be quite different to that obtained when operating over an HF radio channel. Indeed, new techniques may need to be developed in order for the estimator to be successful in its tracking of such fast fading channels.

This chapter describes the outcome of a feasibility study of some new estimation processes [13] and also describes some novel techniques for estimating the response of channels that introduce flat Rayleigh fading with a high fading rate. A further aim of this chapter is to study the feasibility of the simultaneous transmission of two bandlimited 4-QAM (QPSK) signals over two independent, flat Rayleigh fading channels to a single receiver, where both signals occupy the same frequency band. No coding or diversity techniques are employed here to improve the performance, other than differential coding needed to avoid prolonged error bursts. The above situation may occur in cellular mobile radio systems where the base station must carry out a process of estimation and detection on each individual received signal, where these signals originate from different mobiles and therefore, are fading independently [12-16].

The chapter continues in Section 6.2, with a description of the basic assumptions made in the study and Section 6.3 gives the assumed model of the data

transmission system. Section 6.4 describes the estimators that are tested, and gives the theory for a novel estimator known here as the 'unbiased estimator'. Section 6.5 examines the detection techniques that are used, which includes a novel combined detector-estimator. Finally, Section 6.6 gives the results of some computer simulation tests, and compares and assesses the performance of the various systems.

6.2 BASIC ASSUMPTIONS

To restrict the depth of the analysis to within reasonable limits and to keep it as general as possible, a number of idealised assumptions are made, both about the channel and about the digital communication system itself.

Firstly, it should be noted here that the initial objective is to determine the relative performance of the estimators when operating under conditions of fast, flat Rayleigh fading similar to those experienced in cellular land mobile radio [12-16]. However, the fading channel model used in the study is not one that is derived specifically for a cellular land mobile radio channel. Rather, it is the fading channel model derived in Section 2.6, for the generation of a single Rayleigh fading skywave (Figs. 2.6.2 and 2.6.3) albeit with the corresponding changes in the Bessel filter coefficients, which are necessary to produce higher fading rates. This assumption has been made to keep the fading model as general as possible, rather than confine it to the particular spectral shaping obtained in an urban mobile radio environment [17]. It is also assumed that the fading introduced by the model is purely random fading, without any constant Doppler shift. This does not invalidate the results since the maximum short term fading rate in the assumed model is several times the average fading rate, so that any channel estimator capable of tracking the given channel should, if anything, be more successful in tracking a more realistic channel model having the same fading rate but less random fading. Thus, although the estimators are being tested purely under conditions of fast, flat Rayleigh fading, the relative performance should be an adequate indication of their robustness (or otherwise) in an urban mobile radio environment.

In the model of the system, the radio carrier frequency (RF) is assumed to be close to 900 MHz. and each radio channel is assumed to have a bandwidth of 25 kHz., thus making it a narrowband system. With a frequency guard band of 1 kHz. between adjacent radio channels to handle variations in filter characteristics, the available channel bandwidth becomes 24kHz. With a 4-level QAM signal and full raised-cosine spectral shaping of the recovered baseband modulating waveform at the receiver, a transmission rate of 24 kbit/s can be achieved over each channel [18]. Allowing 20% redundancy in

this signal for synchronising, training and control purposes, a useful transmission rate of 19.2 kbit/s is achieved.

As mentioned before, the other aim of the investigation is to study the feasibility of receiving two 4-level QAM signals, where these are received over two independently fading paths (as would be the case when transmitted from two different mobiles to the same base station), but occupy the same frequency band. This means that both signals are assigned the same radio channel of 25kHz., thus increasing the bandwidth efficiency of the system. The useful transmission rate over the given frequency band is now increased from 19.2 to 38.4 kbit/s.

6.3 MODEL OF THE SYSTEM

The model of the data transmission system is shown in Fig. 6.3.1, where all signals are baseband and complex valued, except $\alpha_{0,i}$, $\alpha_{1,i}$, $\alpha'_{0,i-n}$ and $\alpha'_{1,i-n}$, which are binary valued. The model is that of a synchronous, serial data transmission system employing two 4-level QAM signals, each of which is transmitted over a fading path that is independent of the path of the other signal. The signal element rate is assumed to be 12 kbaud for each signal, resulting of course, in an information rate of 24 kbit/s, per fading path. When both signals are received in the given frequency band, the total information rate is 48 kbit/s.

The transmitted information is in the form of two sequences of binary digits $\{\alpha_{0,i}\}$ and $\{\alpha_{1,i}\}$, where the digits in each sequence are statistically independent and equally likely to have either of the binary values 0 or 1. The sequences $\{\alpha_{0,i}\}$ and $\{\alpha_{1,i}\}$ are fed to two identical differential encoders, which give at their outputs, the corresponding differentially encoded signals. The differential encoding of each stream of binary digits is performed exactly as given in Appendix B, where the outputs of the two differential encoders are two corresponding sequences of data symbol values $\{s_{0,i}\}$ and $\{s_{1,i}\}$. These data symbol values are assumed to be in the form of impulses, $\{s_{0,i}\,\delta(t\text{-}iT)\}$ and $\{s_{1,i}\,\delta(t\text{-}iT)\}$, occuring at 12 kbaud, i.e. T = 1/12000 secs. The data symbols $\{s_{0,i}\}$ and $\{s_{1,i}\}$ are statistically independent and equally likely to have any one of four given values, which may be either $\pm1\pm j$, or else, ±1 or $\pm j$, where $j = \sqrt{-1}$. Moreover, it is assumed that $s_{0,i} = s_{1,i} = 0$ for $i \leq 0$, so that both $s_{0,i}$ and $s_{1,i}$ are the i^{th} transmitted data symbol values.

The two signal streams, $\{s_{0,i}\,\delta(t - iT)\}$ and $\{s_{1,i}\,\delta(i - iT)\}$, are fed to two

lowpass filters Z_o and Z_1 , whose transfer functions are identical and root raised-cosine in shape. The signals at the outputs of Z_0 and Z_1, are fed to the transmission paths TX_0 and TX_1 , respectively. Each transmission path is a linear baseband channel that introduces the baseband equivalent of flat Rayleigh fading. Each transmission path could thus be considered to comprise a single Rayleigh fading skywave, whose generation is as depicted in Figs. 2.6.2 and 2.6.3, and explained in Section 2.6, and whose baseband equivalent model is as derived in Section 3.3, albeit now with only one skywave. The frequency spread introduced by each transmission path is 125 Hz. and the coefficients of the Bessel filter (Fig. 2.6.3) that gives rise to this frequency spread, are given in Appendix A. The sampling rate of the Bessel filter used in this case is 1200 Hz. and linear interpolation has been employed to generate samples at 12000 per second. The value of $(G_D)^{-1}$ (Fig. A.1) is 17.4392. A frequency spread of 125 Hz. results in a fade rate of approximately 92 fades per second (eqns. 2.6.2 and 2.6.3). This is a useful, but rather idealised model of the fading likely to be introduced into a 900 MHz. carrier in an urban environment [15-17].

Stationary, zero mean, complex valued, white Gaussian noise w(t), is added to the sum of the two fading signals, as shown in Fig. 6.3.1. The noise waveform w(t), in the baseband equivalent model, is derived from a real valued noise function n(t), which is a Gaussian random process with zero mean and a two sided power spectral density of $N_o/2$ (see Section 3.2, Figs. 3.2.1 and 3.2.7). The resultant noisy and fading signal is next fed through a lowpass filter Z_3 , to give the bandlimited noisy waveform r(t). The transfer function of Z_3 is assumed to be root raised-cosine in shape.

The received waveform r(t), is now sampled once per data symbol at $t = iT$, to give the corresponding sequence of received samples $\{r_i\}$. Clearly, since the resultant transfer function of the transmitter and receiver filters is raised-cosine in shape, the appropriate phase in the sampling instants would ensure that there is no intersymbol interference in the received samples $\{r_i\}$, where $r_i = r(iT)$, the delay in transmission being neglected here [18]. Thus, the received sample at time $t = iT$ is

$$r_i = s_{0,i}y_{0,i} + s_{1,i}y_{1,i} + w_i \qquad (6.3.1)$$

where r_i , $y_{0,i}$, $y_{1,i}$ and w_i are complex valued. The lowpass filter Z_3 is such that the real and imaginary parts of the noise components $\{w_i\}$, are statistically independent Gaussian random variables with zero mean and a fixed variance σ^2 [18]. The quantities $y_{0,i}$ and $y_{1,i}$ may vary quite rapidly with i, and each represents the attenuation and phase change

introduced into the corresponding signal by the transmission path. The average transmitted energy per bit is E_b at the input to each transmission path. These transmission paths do not introduce, on average, any gain or loss into the data signal, so that the average energy per bit at the output of each transmission path is also E_b.

On receipt of the sample r_i at t = iT, the detector uses the predictions $y'_{0,i,i-1}$ and $y'_{1,i,i-1}$, that were formed of $y_{0,i}$ and $y_{1,i}$, respectively, at t = (i-1)T by the estimator, to determine the detected data symbol values $s'_{0,i-n}$ and $s'_{1,i-n}$. Clearly, the detector introduces a delay of n sampling intervals (in the case of simple detectors, n = 0). The two detected data symbols $s'_{0,i-n}$ and $s'_{1,i-n}$, are then passed through two identical differential decoders, which give at their outputs, the detected binary information digits $\alpha'_{0,i-n}$ and $\alpha'_{1,i-n}$. Differential decoding is the inverse operation of differential encoding, and is fully described in Appendix B, for the given system.

In addition to $s'_{0,i-n}$ and $s'_{1,i-n}$, the detector also provides two "early" detected data symbols $s'_{0,i}$ and $s'_{1,i}$, for use in the channel estimator. The channel estimator uses the values of r_i, $s'_{0,i}$ and $s'_{1,i}$, to form the predictions $y'_{0,i+1,i}$ and $y'_{1,i+1,i}$, of the values $y_{0,i+1}$ and $y_{1,i+1}$, that are needed by the detector at the next sampling instant t = (i+1)T. The process continues this way.

When it is assumed that only one data signal is received in the given frequency band, the second of the two transmission paths is omitted from Fig. 6.3.1 and eqn. 6.3.1 becomes

$$r_i = s_{0,i} y_{0,i} + w_i \qquad (6.3.2)$$

It is also assumed in every case, that at the start of the transmission of any message, the receiver has exact prior knowledge of $y_{0,i}$ and $y_{1,i}$, which corresponds to the ideal practical case where a known training signal is transmitted prior to the data, to enable the receiver to obtain an accurate initial estimate of the channel.

6.4 THE ESTIMATORS

The estimators developed in this work are derivatives of the gradient estimator[5] which uses the least mean square (LMS) algorithm [1] for the adjustment of

the tap gains . As such, the operation of the gradient estimator is given first, and this is followed by a description of the operation of the estimators used in this work.

6.4.1 The Gradient Estimator

In order to facilitate the understanding of the estimators used in this work, it is first necessary to explain the operation of the gradient estimator (which was developed by Magee and Proakis in 1973 [5]), when configured to estimate the responses $y_{0,i}$ and $y_{1,i}$, of two independently fading channels, when flat Rayleigh fading is assumed to occur on both channels.

The gradient estimator for the above case is shown in Fig. 6.4.1, where the signals shown occur at the time instant t = iT. The tap gains of the estimator give the estimated values of the sampled impulse responses of the two channels. These tap gains are adjusted by the stochastic gradient algorithm (which is derived from the steepest descent algorithm [4,19]), in such a way as to minimise the mean square error between the received samples $\{r_i\}$, and the corresponding values $\{r'_i\}$, at the output of the estimator, which are estimates of the received sample. The adjustment of the two tap gains after the receipt of the sample r_i , leads to the two tap gains forming the stored estimates $y'_{0,i}$ and $y'_{1,i}$, of $y_{0,i}$ and $y_{1,i}$.

The channel estimator operates as follows. At t = iT, the received sample r_i and the detected data symbols symbols $s'_{0,i}$ and $s'_{1,i}$, are fed to the estimator. The symbols $s'_{0,i}$ and $s'_{1,i}$ are first multiplied by the corresponding tap gains ($y'_{0,i-1}$ and $y'_{1,i-1}$, respectively), and the products are added to give an estimate r'_i , of the received sample r_i . The tap gains are, of course, the estimated responses of the two channels at t = (i-1)T. Thus,

$$r'_i = s_{0,i}y_{0,i-1} + s_{1,i}y_{1,i-1} \qquad (6.4.1)$$

The error, e_i , in the estimate is taken as,

$$e_i = r_i - r'_i \qquad (6.4.2)$$

where r_i is as given in eqn. 6.3.1. This error signal is multiplied by a small positive

quantity b, to give the signal be_i . The signal be_i now multiplies the complex conjugates of $s'_{0,i}$ and $s'_{1,i}$ (i.e., $(s'_{0,i})^*$ and $(s'_{1,i})^*$), and the respective products are added to the corresponding tap gains to give the new stored estimates $y'_{0,i}$ and $y'_{1,i}$, of the sampled impulse response of the two channels. Thus, the estimates of the sampled impulse response of the two channels at t = iT are,

$$y'_{0,i} = y'_{0,i-1} + b\,e_i(s_{0,i})^* \qquad (6.4.3)$$

and

$$y'_{1,i} = y'_{1,i-1} + b\,e_i(s_{1,i})^* \qquad (6.4.4)$$

Eqns. 6.4.3 and 6.4.4 can be shown to be derived from the steepest descent algorithm for adjusting the tap gains of the channel estimator [1-5].

The above algorithm is widely known as the least mean square (LMS) algorithm [1]. The quantity b is known as the step size, and it controls the speed of convergence and the accuracy of the estimate. The smaller the step size, the smaller is the effect of additive noise on the channel estimates (and hence, the more accurate the channel estimates), but the slower is the rate of response of $y'_{0,i}$ and $y'_{1,i}$, to changes in $y_{0,i}$ and $y_{1,i}$.

6.4.2 The Modified Gradient Estimator

The first of the estimators studied in this work is a modification of the gradient estimator described in Section 6.4.1, whereby a least squares fading memory predictor is used [7,20] to give two predictions $y'_{0,i+1,i}$ and $y'_{1,i+1,i}$, at t = iT, of the sampled impulse responses $y_{0,i+1}$ and $y_{1,i+1}$, of the two channels at t = (i+1)T.

The operation of the estimator is as follows. As before, an estimate, r'_i , of the received signal r_i , is formed at t = iT as,

$$r'_i = s'_{0,i}y'_{0,i,i-1} + s'_{1,i}y'_{1,i,i-1} \qquad (6.4.5)$$

Notice that $y'_{0,i,i-1}$ and $y'_{1,i,i-1}$ are predictions of $y_{0,i}$ and $y_{1,i}$, respectively, that were

obtained at t =(i-1)T. The error in the estimate of the received signal is then formed, and is given by,

$$e_i = r_i - r_i' \tag{6.4.6}$$

where r_i is as given in eqn. 6.3.1. The updated estimates of $y_{0,i}$ and $y_{1,i}$ are next obtained, and are given by,

$$y'_{0,i} = y'_{0,i,i-1} + b\, e_i (s'_{0,i})^* \tag{6.4.7}$$

and

$$y'_{1,i} = y'_{1,i,i-1} + b\, e_i (s'_{1,i})^* \tag{6.4.8}$$

where b is a small positive quantity and $(s'_{0,i})^*$ and $(s'_{1,i})^*$ are the complex conjugates of $s'_{0,i}$ and $s'_{1,i}$, respectively. The errors in the predictions $y'_{0,i,i-1}$ and $y'_{1,i,i-1}$ are then taken to be

$$\varepsilon_{0,i} = y'_{0,i} - y'_{0,i,i-1} = b\, e_i (s'_{0,i})^* \tag{6.4.9}$$

and

$$\varepsilon_{1,i} = y'_{1,i} - y'_{1,i,i-1} = b\, e_i (s'_{1,i})^* \tag{6.4.10}$$

These errors are then fed to the appropriate least squares fading memory polynomial filter (degree 0, 1 or 2) [20] to give the predictions $y'_{0,i+1,i}$ and $y'_{1,i+1,i}$, that are needed for the detection process at t = (i+1)T.

Table 6.4.1 gives the relationships of the degree 0, 1 and 2 least squares fading memory polynomial filters for obtaining the one-step and n-step predictions $y'_{0,i+1,i}$ and $y'_{0,i+n,i}$, at time t = iT. θ, in Table 6.4.1, is a constant that determines the memory of the filter, while the terms $y''_{0,i+1,i}$ and $y'''_{0,i+1,i}$ are functions of the first and second derivatives of $y'_{0,i+1,i}$ with respect to time, and are considered in detail elsewhere [20]. Further consideration of these is not necessary for, and indeed is beyond the scope of, the work in

this thesis. Relationships exactly corresponding to those in Table 6.4.1 also hold for $y'_{1,i+1,i}$, $y''_{1,i+1,i}$ and $y'''_{1,i+1,i}$.

When there is only one received signal, eqn. 6.4.5 becomes

$$r'_i = s'_{0,i} y'_{0,i,i-1} \qquad (6.4.11)$$

and in eqn. 6.4.6, r_i is now (see eqn. 6.3.2)

$$r_i = s_{0,i} y_{0,i} + w_i$$

Clearly, eqns. 6.4.8 and 6.4.10 are redundant, but the system otherwise operates as previously described, to determine $y'_{0,i+1,i}$.

6.4.3 The Unbiased Estimator

It can be seen from eqns. 6.4.9 and 6.4.10 that two updated estimates $y'_{0,i}$ and $y'_{1,i}$, are used in the calculation of the prediction errors $\varepsilon_{0,i}$ and $\varepsilon_{1,i}$, in the predictions $y'_{0,i,i-1}$ and $y'_{1,i,i-1}$. However, in the original least squares fading memory algorithm [20], direct measurements of $y_{0,i}$ and $y_{1,i}$ that are independent of the predictions, are assumed to be used in the calculation of the respective prediction errors $\varepsilon_{0,i}$ and $\varepsilon_{1,i}$, that are fed to the least squares fading memory predictor. The reason for this is that if the updated estimates $y'_{0,i}$ and $y'_{1,i}$ are used instead of direct measurements to calculate prediction errors, then, the system may be prone to instability (especially in the case of degree-2 and higher order predictions), because the updated estimates are themselves functions of the predictions (eqns.6.4.7 and 6.4.8).

This suggests that a useful modification to the algorithm described in Section 6.4.2, may be to determine the estimates $y'_{0,i}$ and $y'_{1,i}$ directly from the received signal in such a way that $y'_{0,i}$ and $y'_{1,i}$ are not dependent on $y'_{0,i,i-1}$ and $y'_{1,i,i-1}$. (Of course, $y'_{0,i}$ and $y'_{1,i}$ will not now be updated estimates). The required result is achieved by means of an algorithm which leads to an estimator which, in the sequel, will be known as the 'unbiased estimator'.

Three different arrangements of the unbiased estimator, based on the degree-0,

1 or 2 polynomial filters, may be considered. These arrangements are appropriate to slow, fast and very fast fading channels, respectively. The resulting estimates of $y_{0,i}$ are referred to here as 'unbiased estimates'. The possible values of $s_{0,i}$ and $s_{1,i}$ are assumed now, to be ± 1 or $\pm j$, as shown in Fig. 6.4.2. This is done purely for convenience since the algorithm requires the calculation of $(s_{0,i})^{-1}$ and $(s_{1,i})^{-1}$, in the estimation process. Also for convenience, all the detected data symbol values are assumed to be correct. The effects of errors in detection are studied later by computer simulation. The algorithms are described for the estimation and prediction of $y_{0,i}$, and the corresponding algorithms for $y_{1,i}$ are obtained from these by interchanging the subscripts 0 and 1.

This section describes the algorithm for the unbiased estimate of a slow fading channel. In order not to impede the flow of the discussion, the algorithms for the unbiased estimate of fast and very fast fading channels, are given in Appendix D.

In the algorithm for the unbiased estimate of a slow fading channel, the unbiased estimator assumes that

$$y_{0,i} = y_{0,i-1} \tag{6.4.12}$$

and

$$y_{1,i} = y_{1,i-1} \tag{6.4.13}$$

For slowly fading channels, eqns. 6.4.12 and 6.4.13 are approximately valid [7]. The key to the estimation of $y_{0,i}$ is the accurate removal of $y_{1,i}$ from the received samples. This can be achieved by operating on the two samples,

$$r_i = s_{0,i}y_{0,i} + s_{1,i}y_{1,i} + w_i \tag{6.4.14}$$

and

$$r_{i-1} = s_{0,i-1}y_{0,i-1} + s_{1,i-1}y_{1,i-1} + w_{i-1} \tag{6.4.15}$$

At time $t = iT$, after the receipt of r_i, the estimator is fed with r_i and the detected values $s'_{0,i}$ and $s'_{1,i}$, of the data symbols $s_{0,i}$ and $s_{1,i}$. Since correct detection is assumed, $s'_{0,i} = s_{0,i}$ and $s'_{1,i} = s_{1,i}$ and so, all data symbols in eqns. 6.4.14 and 6.4.15 are known at the

estimator. Now, from eqn. 6.4.14,

$$s_{1,i}^{-1} r_i = s_{1,i}^{-1} s_{0,i} y_{0,i} + y_{1,i} + s_{1,i}^{-1} w_i \qquad (6.4.16)$$

and from eqn. 6.4.15,

$$s_{1,i-1}^{-1} r_{i-1} = s_{1,i-1}^{-1} s_{0,i-1} y_{0,i-1} + y_{1,i-1} + s_{1,i-1}^{-1} w_{i-1} \qquad (6.4.17)$$

From eqns. 6.4.16 and 6.4.17,

$$\begin{aligned} s_{1,i}^{-1} r_i - s_{1,i-1}^{-1} r_{i-1} = {} & s_{1,i}^{-1} s_{0,i} y_{0,i} - s_{1,i-1}^{-1} s_{0,i-1} y_{0,i-1} \\ & + y_{1,i} - y_{1,i-1} \\ & + s_{1,i}^{-1} w_i - s_{1,i-1}^{-1} w_{i-1} \end{aligned} \qquad (6.4.18)$$

Substituting the values of eqns. 6.4.12 and 6.4.13 in eqn. 6.4.18 gives,

$$\begin{aligned} s_{1,i}^{-1} r_i - s_{1,i-1}^{-1} r_{i-1} \approx {} & (s_{1,i}^{-1} s_{0,i} - s_{1,i-1}^{-1} s_{0,i-1}) y_{0,i} \\ & + s_{1,i}^{-1} w_i - s_{1,i-1}^{-1} w_{i-1} \end{aligned} \qquad (6.4.19)$$

Eqn. 6.4.19 can be expressed as

$$r_{0,i} \approx a_{0,i} y_{0,i} + u_{0,i} \qquad (6.4.20)$$

where

$$r_{0,i} = s_{1,i}^{-1} r_i - s_{1,i-1}^{-1} r_{i-1} \qquad (6.4.21)$$

$$a_{0,i} = s_{1,i}^{-1} s_{0,i} - s_{1,i-1}^{-1} s_{0,i-1} \qquad (6.4.22)$$

and

$$u_{0,i} = s^{-1}_{1,i}\, w_i - s^{-1}_{1,i-1}\, w_{i-1} \qquad (6.4.23)$$

From eqn. 6.4.20 it could be seen that an estimate of $y_{0,i}$ is given by

$$x_{0,i} \approx a^{-1}_{0,i}\, r_{0,i} \approx y_{0,i} + a^{-1}_{0,i}\, u_{0,i} \qquad (6.4.24)$$

$x_{0,i}$ is called an 'unbiased estimate' of $y_{0,i}$. This channel estimate is then compared with the corresponding one-step prediction $y'_{0,i,i-1}$, to give the measured error $(x_{0,i} - y'_{0,i,i-1})$, in th prediction $y'_{0,i,i-1}$.

The variance of $u_{0,i}$ is independent of $s_{0,i}$, $s_{1,i}$, $s_{0,i-1}$ and $s_{1,i-1}$ and therefore it is also independent of $a_{0,i}$. Thus, the variance of $(a_{0,i})^{-1}\, u_{0,i}$ is inversely proportional to $|a_{0,i}|^2$, and hence, the greater the value of $|a_{0,i}|$, the smaller the variance of $(a_{0,i})^{-1}\, u_{0,i}$, and so, the better is the estimate of $y_{0,i}$ that is given by $x_{0,i}$. To make due allowance for the changes in the variance of $(a_{0,i})^{-1}\, u_{0,i}$ with i, and to provide more adjustability when applying the measured prediction error $(x_{0,i} - y'_{0,i,i-1})$ to the least squares fading memory prediction algorithm, the prediction error is now redefined as,

$$\varepsilon_{0,i} = b\,|a_{0,i}|\,(x_{0,i} - y'_{0,i,i-1}) \qquad (6.4.25)$$

where b is, again, a small positive constant, similar to the one that was used in the modified gradient estimator (eqns. 6.4.7 and 6.4.8). The measured error, $\varepsilon_{0,i}$, is now used for the degree-0 polynomial filter in Table 6.4.1, to give the one-step prediction of $y_{0,i+1}$. The corresponding arrangement, with the subscripts 0 and 1 interchanged, gives the prediction of $y_{1,i+1}$. These predictions are used in the next detection process at t = (i+1)T.

Whenever $a_{0,i} = 0$, it could be seen from eqn. 6.4.24 that the algorithm cannot be used to give the unbiased estimate $x_{0,i}$. The procedure now, is to ignore r_i in the estimation and prediction process, and continue instead with r_{i+1} . The detector would now use the two-step predictions, $y'_{0,i+1,i-1}$ and $y'_{1,i+1,i-1}$, for the detection of $s_{0,i+1}$ and $s_{1,i+1}$ from r_{i+1}, while the estimator would evaluate $x_{0,i+1}$ (see eqn. 6.4.24) and the prediction error,

$$\varepsilon_{0,i+1} = b\,|a_{0,i+1}|\,(x_{0,i+1} - y'_{0,i+1,i-1}) \qquad (6.4.26)$$

in which the two-step prediction of $y_{0,i+1}$ is used. The least squares fading memory prediction algorithm (Table 6.4.1) now operates on the appropriate two-step predictions, for time t = (i+1)T, instead of on the corresponding one-step predictions, to give the required one and n-step predictions. In the case of the degree-0 polynomial filter used here, the one and two-step predictions are, of course, the same.

If now, n successive $\{r_{i+h}\}$, (h = 0, 1, ..., n-1), give zero values for the corresponding $\{a_{0,i+h}\}$, the (h+2)-step predictions of the $\{y_{0,i+h+1}\}$ must be used for the detection of the corresponding data symbols, no estimation or prediction processes being carried out during this time. The (n+1)-step prediction of $y_{0,i+n}$ is then used in the evaluation of $\varepsilon_{0,i+n}$. The least squares fading memory prediction algorithm (Table 6.4.1) now operates on the appropriate (n+1)-step predictions, for time t = (i+n)T, instead of on the corresponding one-step predictions. Similar procedures should be carried out for the detection of the $\{s_{1,i+h}\}$ and the estimation of the $\{y_{1,i+h}\}$ when $a_{1,i+h} = 0$, for h = 0,1, ..., n-1 .

The derivations of the unbiased estimates of fast and very fast fading channels are given in Appendix D. Both these derivations follow the same basic approach as that in the analysis just presented. However, the estimation processes themselves are substantially more complex because of the necessity here, to correct for changes in $y_{0,i}$ and $y_{1,i}$, with i .

When only one signal is received, such that

$$r_i = s_{0,i}\,y_{0,i} + w_i \qquad (6.4.27)$$

the unbiased estimate of $y_{0,i}$ becomes

$$x_{0,i} \approx s_{0,i}^{-1}\,r_i \approx y_{0,i} + s_{0,i}^{-1}\,w_i \qquad (6.4.28)$$

and is therefore obtained from just one received sample. The error in the one-step prediction is now taken as

$$\varepsilon_{0,i} = b\,(s_{0,i}^{-1}\,r_i - y'_{0,i,i-1}) \qquad (6.4.29)$$

which is then used in the appropriate prediction algorithm of Table 6.4.1, to give $y'_{0,i+1,i}$. Correct detection of $s_{0,i}$ and $s_{1,i}$ has been assumed in the above analysis.

6.5 THE DETECTORS

6.5.1 The Simple Detector

The simple detector is essentially a threshold level detector. The term 'simple' is deemed appropriate since it distinguishes this detector from a more sophisticated detector to be described in the next section.

At time t = iT, the detector receives in addition to the received sample r_i , the predictions $y'_{0,i,i-1}$ and $y'_{1,i,i-1}$, of $y_{0,i}$ and $y_{1,i}$, that were formed at t = (i-1)T by the channel estimator. The detector uses this information to calculate

$$d_i^2 = | r_i - q_{0,i} y'_{0,i,i-1} - q_{1,i} y'_{1,i,i-1} |^2 \qquad (6.5.1)$$

where |x| is the absolute value of x and $q_{0,i}$ and $q_{1,i}$ take on the possible values of $s_{0,i}$ and $s_{1,i}$. Since $s_{0,i}$ and $s_{1,i}$ are both 4-level signals, there are 16 possible values of $(d_i)^2$ associated with the 16 possible combinations of the values of $s_{0,i}$ and $s_{1,i}$. The detected data symbols $s'_{0,i}$ and $s'_{1,i}$, are now chosen by the detector as the values of $q_{0,i}$ and $q_{1,i}$, respectively, that give rise to the lowest possible $(d_i)^2$, i.e., the possible values of $q_{0,i}$ and $q_{1,i}$ that minimise $(d_i)^2$.

6.5.2 The Combined Detector

As described in Section 6.4, the estimators used in this work rely on correct detection of the data symbols. This suggests that an estimation process that considers several different possible values of each detected data symbol, may give an advantage over an estimation process that considers just the single possible value. The combined detector and estimator achieves this objective by storing a number, say p, of possible transmitted sequences (as in the case of a near maximum likelihood detector - Chapter 4), and having a different estimator associated with each transmitted sequence. This combination of detection and estimation is carried out as follows.

Consider a composite data symbol given by a two-component vector

$$q_i = [q_{0,i} \quad q_{1,i}] \tag{6.5.2}$$

where $q_{0,i}$ and $q_{1,i}$ take on the possible values of $s_{0,i}$ and $s_{1,i}$, respectively. Thus, q_i has 16 different possible values corresponding uniquely to the 16 different possible combinations of $s_{0,i}$ and $s_{1,i}$.

Just prior to the receipt of the sample r_i at time t = iT, the detector holds in store p different n-component vectors $\{Q_{i-1}\}$, where,

$$Q_{i-1} = [q_{i-n} \quad q_{i-n+1} \quad \cdots\cdots \quad q_{i-1}] \tag{6.5.3}$$

Each vector Q_{i-1}, represents a different possible pair of the sequences

$$[s_{0,i-n} \quad s_{0,i-n+1} \quad \cdots\cdots \quad s_{0,i-1}]$$

and

$$[s_{1,i-n} \quad s_{1,i-n+1} \quad \cdots\cdots \quad s_{1,i-1}]$$

Associated with each vector Q_{i-1}, is a cost function which is a measure of the likelihood that the vector is correct, the lower the cost, the higher being the likelihood. This cost function is given by

$$c_{i-1} = \sum_{h=1}^{i-1} \phi^{h-1} \left| r_{i-h} - q_{0,i-h} y'_{0,i-h,i-h-1} - q_{1,i-h} y'_{1,i-h,i-h-1} \right|^2 \tag{6.5.4}$$

where ϕ is a real valued constant in the range 0 to 1, and |x| is the absolute value of the scalar quantity x. It is assumed that

$$q_{0,i} = q_{1,i} = s_{0,i} = s_{1,i} = 0 \quad \text{for} \quad i \leq 0 \tag{6.5.5}$$

The nearer ϕ approaches to zero, the smaller is the effect of earlier costs on c_i , thus reducing the effective memory in c_i .

On receipt of the sample r_i , each of the p vectors $\{Q_{i-1}\}$ is expanded into m (n+1)-component vectors $\{P_i\}$, where,

$$P_i = [\, q_{i-n} \quad q_{i-n+1} \quad \ldots\ldots \quad q_{i-1} \quad q_i \,] \tag{6.5.6}$$

and m is a variable that decreases as the cost of the given Q_{i-1} increases. In each group of m vectors $\{P_i\}$ which were derived from a given vector Q_{i-1} , the first n components $\{q_{i-h}\}$, (h = n, n-1, ..., 1), are as in the original Q_{i-1} , and the last component q_i , takes on m different possible values. When m < 16, the m vectors $\{P_i\}$ originating from any one Q_{i-1} , are those with the smallest costs in the group of 16 possible vectors that can be derived from the given Q_{i-1} . Stored alongside each of the resulting vectors $\{P_i\}$, are their costs,

$$c_i = \phi c_{i-1} + \left| r_i - q_{0,i} y'_{0,i,i-1} - q_{1,i} y'_{1,i,i-1} \right|^2 \tag{6.5.7}$$

where c_{i-1} is the cost of the vector Q_{i-1} , from which the particular vector P_i is derived.

At this stage, the detector holds in store a number of expanded vectors $\{P_i\}$. From these expanded vectors $\{P_i\}$, is now chosen the vector with the smallest cost. Let this chosen vector be denoted P_s . The first component q_{i-n} , of P_s , is then taken as the value which gives the detected values $s'_{0,i-n}$ and $s'_{1,i-n}$, of the data symbols $s_{0,i-n}$ and $s_{1,i-n}$ (see eqn. 6.5.2). Any vector P_i whose first component differs in value from the first component of P_s , is then discarded by assigning to it an arbitrarily high value of cost. From the remaining vectors $\{P_i\}$, (which includes P_s), are selected the p vectors which have the smallest costs. The first component of each of the p selected vectors $\{P_i\}$, is now omitted (without changing its cost), to give the corresponding n-component vectors $\{Q_i\}$, where,

$$Q_i = [\, q_{i-n+1} \quad q_{i-n+2} \quad \ldots\ldots \quad q_i \,] \tag{6.5.8}$$

These p vectors $\{Q_i\}$, are then stored along with their associated costs $\{c_i\}$. The smallest of these costs is now subtracted from each of the p costs, in order to prevent cost overflow. With the p stored vectors $\{Q_i\}$, and their associated costs $\{c_i\}$, the detector is ready for the next detection process, that is, the detection of s_{i-n+1}, after the receipt of r_{i+1}. When ϕ=1, (eqns. 6.5.4 and 6.5.7), the detector just described becomes a direct development of the near maximum likelihood detectors that were discussed in Chapter 4.

In the detector that is used in the simulation tests, p = 4 and m has the values 4, 3, 2 and 1, respectively, for the four $\{Q_{i-1}\}$, when arranged in the order of increasing costs and starting with the lowest cost vector. Thus, on the receipt of r_i, the first, second, third and fourth vectors $\{Q_{i-1}\}$, are expanded into four, three, two and one vectors $\{P_i\}$, respectively. There are now, ten expanded vectors $\{P_i\}$, from which are selected four vectors $\{Q_i\}$, as previously described.

With no intersymbol interference (as is the case here), and a single estimation and prediction process, no advantage would be gained by the arrangement just described, over the simple detector that was given in Section 6.5.1. However, in the system studied here, each of the four vectors $\{Q_{i-1}\}$, is associated with its own separate estimator and predictor, which may operate by any of the methods described in Section 6.4, and which take the received sequences of data symbol values $\{s_{0,h}\}$ and $\{s_{1,h}\}$ to be those given by the corresponding vector Q_{i-1}. Thus, there are four separate estimation and prediction processes operating in parallel. For convenience, eqns. 6.5.4 and 6.5.7 assume the use of a modified gradient estimator (Section 6.4.2), so that the predictions of $y_{0,i}$ and $y_{1,i}$ that are employed, are one-step predictions.

When a vector Q_{i-1} is expanded into m vectors $\{P_i\}$, the same predictions of $y_{0,i}$ and $y_{1,i}$ are used to obtain each of the m vectors $\{P_i\}$, but these predictions normally differ from those associated with any of the other three vectors $\{Q_{i-1}\}$. After the selection of the four vectors $\{Q_i\}$ from the ten vectors $\{P_i\}$, the prediction errors $\varepsilon_{0,i}$ and $\varepsilon_{1,i}$, are evaluated separately for each Q_i. Then, for each of these vectors, $\varepsilon_{0,i}$ is applied to the appropriate prediction algorithm of Table 6.4.1, to give the one-step prediction $y'_{0,i+1,i}$, of $y_{0,i+1}$. Of course, $\varepsilon_{1,i}$ is handled similarly, to give the one-step prediction $y'_{1,i+1,i}$, of $y_{1,i+1}$. Thus, since the four $\{Q_i\}$ are different, so also, in general, are the predictions of $y_{0,i+1}$ and $y_{1,i+1}$ that are associated with the four $\{Q_i\}$. The techniques just described not

only make the estimator more tolerant of errors in the detected data symbols, but they also reduce the possibility of a complete failure in the detection and estimation processes, when these are operating together.

6.6 COMPUTER SIMULATION TESTS AND RESULTS

Before discussing the results of the computer simulation tests that have been carried out to compare the performance of the different estimation and prediction processes, it is necessary first to note the values of the parameters used and the additional assumptions made, in the tests.

The model of the system that is used in the tests is that given in Section 6.3. The only exception to this is that ideal lowpass filtering with a bandwidth of 1/2T Hz. is assumed for the lowpass filters so that the signal is in fact transmitted over the channel at the Nyquist rate. This is done in order that only one sample per signal element is involved in the processing of the signal, thereby substantially reducing the computer time needed for the tests. No significant change in performance should be introduced by this modification.

In all tests, the average transmitted signal energy per bit is,

$$E_b = \frac{1}{2} |s_{0,i}|^2 = \frac{1}{2} |s_{1,i}|^2 \qquad (6.6.1)$$

and the mean square values of $y_{0,i}$ and $y_{1,i}$ are both equal to unity. The signal-to-noise ratio is ψ dB, where,

$$\psi = 10 \log_{10} \left(\frac{E_b}{\frac{1}{2} N_0} \right) \qquad (6.6.2)$$

For the filters assumed in the model, it is easily shown (see Appendix C - eqn.21 onwards) that

$$\frac{1}{2} N_0 = \sigma^2 \qquad (6.6.3)$$

where σ^2 is the variance of the real or imaginary part of the noise samples $\{w_i\}$ in eqn.

6.3.1. Thus, from eqns. 6.6.2 and 6.6.3,

$$\psi = 10 \log_{10} \left(\frac{E_b}{\sigma^2} \right) \qquad (6.6.4)$$

It could be seen that full allowance is made here for the transmitted signal level.

Figs. 6.6.1 and 6.6.2 show the signal level fading and Figs. 6.6.3 and 6.6.4 show the phase variations introduced by the two transmission paths TX_0 and TX_1, respectively. These fading sequences have been chosen to be typically 'worst-case' sequences for the given parameters of the channel. In the rest of the figures, the nomenclature given below has been followed.

LSFM - least squares fading memory

MGE - modified gradient estimator

UE - unbiased estimator

SD - simple detector

CD - combined detector

Figs. (6.6.5 - 6.6.8) show the performance of the modified gradient estimator in various situations, and Figs. (6.6.9 - 6.6.13) show the performance of the unbiased estimator in the same situations. The parameter b is that given in Section 6.3 and 6.4 for the two different estimators, and the parameter θ is that given in Table 6.4.1. The measurements of the mean square error are all carried out on the signal received over the first transmission path TX_0, which introduces the more severe fading of the two. The measurements of the mean square error in a prediction of $y_{0,i}$, are expressed in decibels relative to unity, and are given by

$$\eta = 10 \log_{10} \left(\frac{1}{21600} \sum_{i=2401}^{24000} | y_{0,i} - y'_{0,i,i-h} |^2 \right) \qquad (6.6.5)$$

where h = 1 or 2, denoting, respectively, the one-step or two-step predictions. The first 2400 $\{y'_{0,i,i-h}\}$ are ignored here in order to eliminate the transient effects that sometimes

occur at the start of operation [20].

Figs. (6.6.13 - 6.6.20) show the performance of the modified gradient estimator and unbiased estimator in similar situations to those in Figs. (6.6.5 - 6.6.12), albeit with the performance measure now being a normalised version, η_N, of the prediction error, where,

$$\eta_N = 10 \log_{10} \left(\frac{1}{21600} \sum_{i=2401}^{24000} \left\{ \frac{|y_{0,i} - y_{0,i,i-h}|}{|y_{0,i}|} \right\}^2 \right) \qquad (6.6.6)$$

The use of this measurement is justified as follows. A possible weakness of the error measurement in eqn. 6.6.5 is the fact that a given error in an estimate of $y_{0,i}$, has an increasingly serious effect on the detection of $s_{0,i}$, as the magnitude of $y_{0,i}$ decreases. Furthermore, most errors in the detection of $s_{0,i}$ occur when the magnitude of $y_{0,i}$ is small, so that the smaller the magnitude of $y_{0,i}$, the more important it becomes that an accurate estimate of $y_{0,i}$ is used. These considerations suggest the use of the error measurement η_N (eqn. 6.6.6), where the error in the estimate of $y_{0,i}$ is taken as the absolute value of the error itself, divided by the magnitude of $y_{0,i}$. The error η_N is known in the sequel, as the mean square normalised error. Whereas the mean square error is influenced mainly by the errors in the channel estimate at high signal levels, the mean square normalised error is influenced mainly by the errors at low signal levels, that is, during fades.

Fig. 6.6.21 shows that performance of a hybrid of the two unbiased estimators described in Appendix D (Section D.1 and D.2). An unbiased estimate of $y_{0,i}$ is here taken as $x_{0,i}$ in eqn. D.18, but it is used in the degree-2 least squares fading memory prediction algorithm in Table 6.4.1. Thus, the unbiased estimate of $y_{0,i}$ is that appropriate to a fast fading channel, whereas the predictions of $y_{0,i+1}$ and $y_{0,i+2}$ are those appropriate to a very fast fading channel. Similar operations are carried out to estimate $y_{1,i}$ and predict $y_{1,i+1}$ and $y_{1,i+2}$.

In Figs. (6.6.5 - 6.6.21), correct detection of all data symbols is assumed, so that only the effects of fading and noise are considered. This gives the performance of the estimators when a known signal is transmitted. Also, the values of b and θ shown on the diagrams are their optimum (approximately) values, where the optimization is in the sense that the particular values of b and θ give rise to the minimum (approximately) mean square

error or minimum (approximately) mean square normalised error.

Figs. (6.6.22 - 6.6.25) show the performance of various combinations of detector and estimator operating in various situations. The parameter ϕ is that given in eqns. 6.5.4 and 6.5.7. Again, the parameters b, θ and ϕ are approximately optimised, but now the optimization is carried out to minimize the bit error rate rather than the mean square prediction error or mean square normalised prediction error. The bit error rate is the error rate in the detected stream of binary digits $\{\alpha'_{0,i}\}$, at the output of the differential decoder in Fig. 6.3.1. Tests have shown that differential coding is essential here to avoid prolonged error bursts being introduced by the deeper fades.

Fig. 6.6.22 assumes perfect channel estimation at the receiver. In Figs. (6.6.23 - 6.6.25), the detected data symbols are used in adjusting the estimators, so that full account is taken of the effects of errors in the data symbols, on the adjustment of the estimators. All the estimators in Figs. (6.6.23 - 6.6.25) are used with the degree-1 least squares fading memory prediction algorithm that is given in Table 6.4.1.

Fig. 6.6.24 assumes that, after every 50 received samples, the predictions of $y_{0,i}$ and $y_{1,i}$ are corrected by some ideal retraining process. Fig. 6.6.25 assumes that, not only the predictions of $y_{0,i}$ and $y_{1,i}$, but also the predictions of their rates of change are corrected every 50 samples. In practice, exact correction of these quantities is not possible, so that the results given in these figures are 'lower bounds' to the bit error rates actually obtained for the given cases. Tests with two received signals and no retraining, where the receiver has no prior knowledge of $y_{0,i}$ and $y_{1,i}$, and so must estimate them, have shown that the system fails completely, regardless of the type of detector (simple or combined) used.

It is clear from Figs. 6.6.1 and 6.6.2 that error free performance is unlikely to be obtained in the presence of any significant noise, due to the very high attenuation introduced by the deeper fades. The estimators are thus being tested under much more severe conditions than would be the case with a more realistic model of the channels. Figs. 6.6.3 and 6.6.4 confirm that the phase of the fading signal varies, as expected, between $-\pi$ and $+\pi$. The distribution functions for the amplitude and phase were obtained in the verification of the channel model in Chapter 2, Section 2.7 (albeit for different channel parameters) and they confirmed that the amplitude of the fading signal follows a Rayleigh distribution and that the phase of the fading signal follows a uniform distribution.

Figs. (6.6.5 - 6.6.9) show the surprisingly good estimates that can be achieved

of the channel with the modified gradient estimator and least squares fading memory prediction, when one signal is received, bearing in mind that correct detection is assumed but no retraining is performed. There is an increase of some 11.5 dB in the mean square prediction error at high signal-to-noise ratios, when two signals are used in place of one. Furthermore, the degree-2 least squares fading memory predictor has a performance similar to that of the degree-1 predictor at high signal-to-noise ratios and for both one and two signals, albeit with a slight advantage in the latter case. However, since the optimum value of b varies with the signal-to-noise ratio, and since in practice a fixed value of b will have to be used in accordance with the signal-to-noise ratio at which optimum performance is most needed, the average performance here is likely to fall somewhat below optimum.

It can be shown theoretically (Appendix E) that, when one signal is received, the unbiased estimator becomes equivalent to the modified gradient estimator, so that the two estimators should now have the same performance. This is indeed confirmed (at least within the limits of experimental accuracy) by Figs. 6.6.9 and 6.6.10, which show the performance of the unbiased estimator when one signal is received. These two figures should be compared with Figs. 6.6.5 and 6.6.6, i.e., the corresponding figures for the modified gradient estimator. However, when two signals are received in the given frequency band, the performance of the unbiased estimator as shown in Figs. 6.6.11 and 6.6.12 is, on average, inferior to that of the modified gradient estimator (Figs. 6.6.7 and 6.6.8), especially at low signal-to-noise ratios.

Figs. (6.6.13 - 6.6.20) show the performance of the two estimators when the measure of performance is the mean square normalised prediction error. These figures include all cases of interest and are thus equivalent to Figs. (6.6.5 - 6.6.12), given the difference in the performance measure. Examination of the results in Figs. (6.6.13 - 6.6.20), and comparison with the corresponding results in Figs. (6.6.5 - 6.6.12) reveal an unexpected consistency between the two measures of performance. The mean square error shows only a small overall advantage for the modified gradient estimator over the unbiased estimator, whereas the mean square normalised error shows a somewhat greater advantage. However, the overall assessment of the different systems tested is not significantly affected by the performance measure used. This is an encouraging result since it further supports the use of the mean square error as a measure of performance of an estimator.

Fig. 6.6.21 compares the performance of the unbiased estimator in the hybrid arrangement that was described earlier, with the performance of the unbiased estimator in its original form from which the hybrid arrangement is derived. Clearly, the hybrid system shows a superior performance. However, its overall performance is still inferior to that of

the corresponding modified gradient estimator with degree-2 least squares fading memory prediction (cf. Fig. 6.6.8).

Fig. 6.6.22 shows that, with perfect channel estimation, there is, on average, a loss in tolerance to additive white Gaussian noise of nearly 5 dB, when two signals are transmitted in place of one, and at the same average transmitted energy per bit. This is similar to the loss experienced when a 16-QAM signal is transmitted in place of a 4-QAM signal, again at the same average transmitted energy per bit. Thus, the two fading channels TX_0 and TX_1 (Fig . 6.3.1) are themselves multiplexing the two signals such that these can be transmitted over the same frequency band without seriously interfering with each other, just so long as sufficiently accurate channel estimates can be obtained in each case.

Fig. 6.6.23 shows that, over the range of bit error rates from 10^{-1} to 10^{-3}, the simple detector and modified gradient estimator, when operating together, lose between 0.5 and 1.5 dB in tolerance to additive white Gaussian noise, relative to the case where the simple detector is used with a perfect estimator. Both the combined detector and modified gradient estimator, and the combined detector and unbiased estimator, lose between 0.5 and 1.0 dB, under the same conditions. These results are very encouraging, bearing in mind that there is no retraining of the channel estimator. At the higher signal-to-noise ratios a combined detector and estimator gains a significant advantage over the simple detector and estimator, and the system with the best overall performance is the combined detector and modified gradient estimator.

Figs. 6.6.24 and 6.6.25 show that, when two signals are received, even with regular retraining of the channel estimate (and its rate of change, in the case of Fig. 6.6.25), there is an unacceptably high bit error rate at high signal-to-noise ratios. However, a useful advantage is gained here by the combined detector and estimator, over the simple detector and estimator. No tests have been carried out where the combined detector is accompanied by the unbiased estimator (rather than a modified gradient estimator), largely due to the high complexity of the unbiased estimator algorithms, when designed for operation with two signals (Appendix D).

Although it is more than likely that a further useful improvement in performance can be achieved through appropriate modifications to the systems described here, it seems improbable that a satisfactory performance will be obtained with two signals, so long as the depth of the fades can exceed 20 dB. The reason for this is that when one of the two signals is attenuated by, say, 20 dB relative to the other, so that $|y_{1,i}| = 10\,|y_{0,i}|$, then, an error of -20 dB in the estimate of $y_{1,i}$ becomes an error of 0 dB relative to the smaller

signal $s_{0,i}\, y_{0,i}$, and can therefore of itself introduce an error in the detection of the smaller signal. Moreover, a similar effect can occur when the two signals have very similar levels and one signal nearly cancels the other. Thus, when there are two signals received in the given frequency band, a much greater accuracy is required in the estimate of the channel.

It has been shown [21] that, when two signals are received over the given frequency band, the above mentioned problems could be alleviated by an appropriate arrangement of space diversity, together with the combined detector and modified gradient estimator. The two antennas reduce the depth of fading to a level that can be handled by the combined detector and estimator, with an acceptable small loss in tolerance to noise relative to the corresponding system that employs perfect channel estimation.

Thus, when only one signal is received in the given frequency band, the preferred system is the combined detector and gradient estimator, which gives a tolerance to noise quite close to that obtained with perfect channel estimation. No retraining of the channel estimator is needed here. In the presence of two signals however, it is necessary to use some sort of diversity scheme, along with regular retraining of the channel estimator.

References - Chapter 6

1. **Haykin, S.,** 'Adaptive filter theory', Prentice Hall, 1986.

2. **Proakis, J.G.,** 'Digital communications', McGraw Hill, 1983.

3. **Clark, A.P.,** 'Equalizers for digital modems', Pentech Press, 1985.

4. **Qureshi, S.U.H,** 'Adaptive equalization', IEEE Proceedings, vol. 73, no.9, pp. 1349 - 1387, September 1985.

5. **Magee, F.R. and Proakis, J.G,** 'Adaptive maximum likelihood sequence estimation for digitial signalling in the presense of intersymbol interfreence', IEEE Trans. on Information Theory, vol. IT-19, pp. 120-124, January 1973.

6. **Clark, A.P., Kwong, C.P. and McVerry, F.,** 'Estimation of the sampled impulse response of a channel', Signal Processing, vol. 2, no.1, pp. 39 - 53, January 1980.

7. **Clark, A.P. and McVerry, F.,** 'Channel estimation for an HF radio link', IEE Proc., Park F, vol. 128, pp. 33 - 42, February 1981.

8. **Clark, A.P. and McVerry, F.,** 'Performance of 2400 bit/s serial and parallel modems over an HF channel simulator', IERE Conf. Proc. no.49, pp.167 - 179, Loughborough, U.K., April 1981.

9. **Clark, A.P. and McVerry, F.,** 'Improved channel estimator for an HF radio link', Signal Processing, no. 5, pp. 241 - 255, May 1983.

10. **Clark, A.P., Nadji, H.Y. and McVerry, F.,** 'Performance of a 9600 bit/s serial modem over a model of an HF radio link', IEE Conf. Publication no.224, pp. 151 - 155, September 1983.

11. **Hariharan, S.,** 'Channel estimators for HF radio links', Ph.D. thesis, Loughborough University of Technology, 1988.

12. **Young, W.R,** 'Advanced mobile phone service: Introduction, background and

objectives', Bell Systems Technical Journal, vol. 58, pp. 1 - 14, January 1979.

13. **Clark, A.P. and Jayasinghe, S.G.,** 'Channel estimation for land mobile radio links', IEE Proc., Part F, vol. 134, no.4, July 1987.

14. **MacDonald, V.H.,** 'Advanced mobile phone service: The cellular concept', Bell Systems Technical Journal, vol. 58, pp. 15 - 41, January 1979.

15. **Arregondo, G.A., Feggeler, J.C and Smith, J.I.,** 'Advanced mobile phone service: Voice and data transmission', ibid, pp. 97 - 122, January 1979.

16. **Clark, A.P.,** 'Digital modems for land mobile radio', IEE Proc, Part F, vol. 132, no. 5, pp. 348 - 362, 1985.

17. **Lee, W.C.Y,** 'Mobile communcations engineering', McGraw Hill, 1982.

18. **Clark, A.P.,** 'Principles of digital data transmission', Pentech Press, 1983.

19. **Widrow, B and Hoff, M.E. Jr**, 'Adaptive switching circuits', IRE WESCON Conv. Rec., Pt .4, pp. 563 - 587, 1960.

20. **Morrison, N.,** 'Introduction to sequential smoothing and prediction', McGraw Hill, 1969.

21. **Brent, J.B.,** 'Coherent detection of QAM signals in land mobile radio', Ph.D. thesis, Loughborough University of Technology, 1988.

Degree of polynomial	One-step and n-step predictions at time $t = iT$
0	$\dot{y}_{0,i+1,i} = \dot{y}_{0,i,i-1} + (1-\theta)\,\varepsilon_{0,i}$ $\dot{y}_{0,i+n,i} = \dot{y}_{0,i+1,i}$
1	$\ddot{y}_{0,i+1,i} = \ddot{y}_{0,i,i-1} + (1-\theta)^2\,\varepsilon_{0,i}$ $\ddot{y}_{0,i+n,i} = \ddot{y}_{0,i+1,i}$ $\dot{y}_{0,i+1,i} = \dot{y}_{0,i,i-1} + \ddot{y}_{0,i+1,i} + (1-\theta^2)\,\varepsilon_{0,i}$ $\dot{y}_{0,i+n,i} = \dot{y}_{0,i+1,i} + (n-1)\,\ddot{y}_{0,i+1,i}$
2	$\dddot{y}_{0,i+1,i} = \dddot{y}_{0,i,i-1} + 0.5\,(1-\theta)^3\,\varepsilon_{0,i}$ $\dddot{y}_{0,i+n,i} = \dddot{y}_{0,i+1,i}$ $\ddot{y}_{0,i+1,i} = \ddot{y}_{0,i,i-1} + 2\,\dddot{y}_{0,i+1,i} + 1.5\,(1-\theta)^2(1+\theta)\,\varepsilon_{0,i}$ $\ddot{y}_{0,i+n,i} = \ddot{y}_{0,i+1,i} + 2\,(n-1)\,\dddot{y}_{0,i+1,i}$ $\dot{y}_{0,i+1,i} = \dot{y}_{0,i,i-1} + \ddot{y}_{0,i+1,i} - \dddot{y}_{0,i+1,i} + (1-\theta^3)\,\varepsilon_{0,i}$ $\dot{y}_{0,i+n,i} = \dot{y}_{0,i+1,i} + (n-1)\,\ddot{y}_{0,i+1,i} + (n-1)^2\,\dddot{y}_{0,i+1,i}$

Table 6.4.1 Least squares fading memory prediction using a polynomial filter

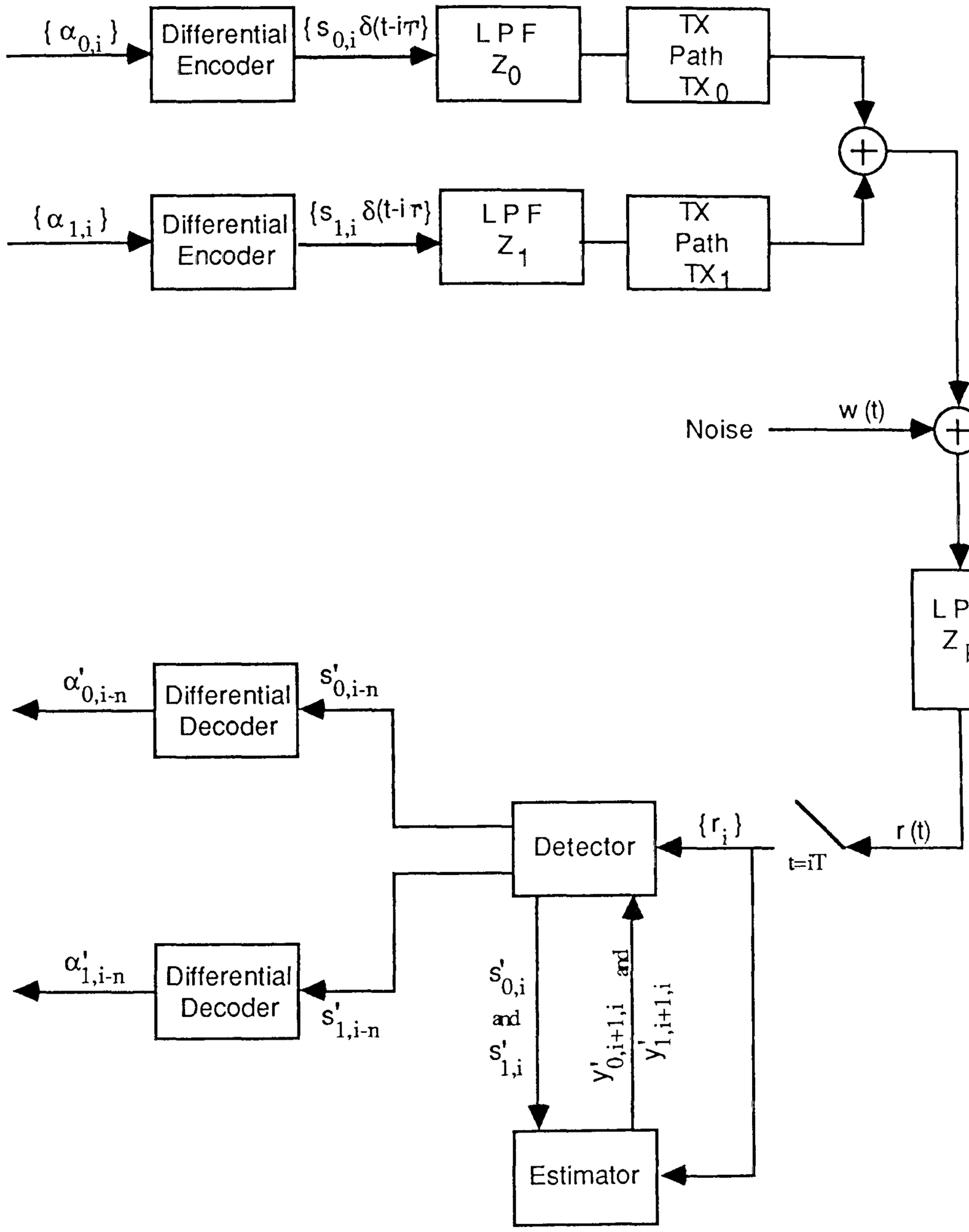


Fig. 6.3.1 Model of data transmission system

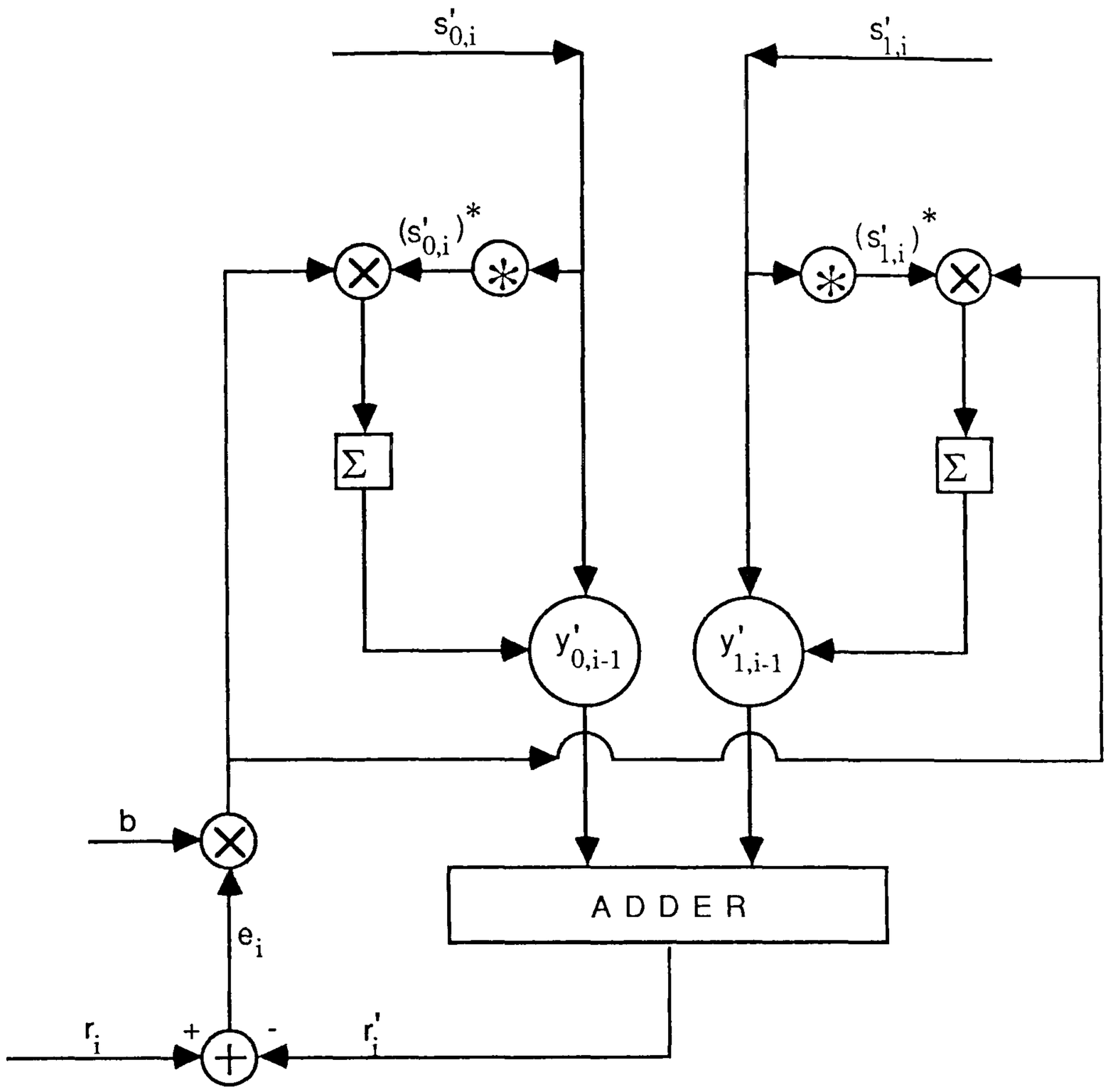


Fig. 6.4.1 Gradient estimator for two independently fading channels

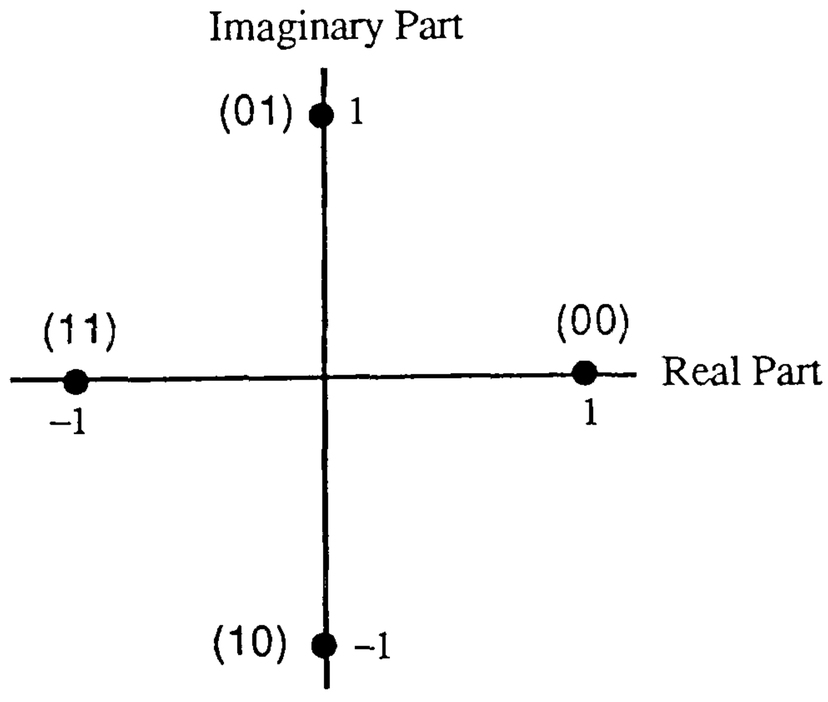


Fig. 6.4.2 Possible values of $s_{0,i}$ and $s_{1,i}$ as used in the unbiased estimator

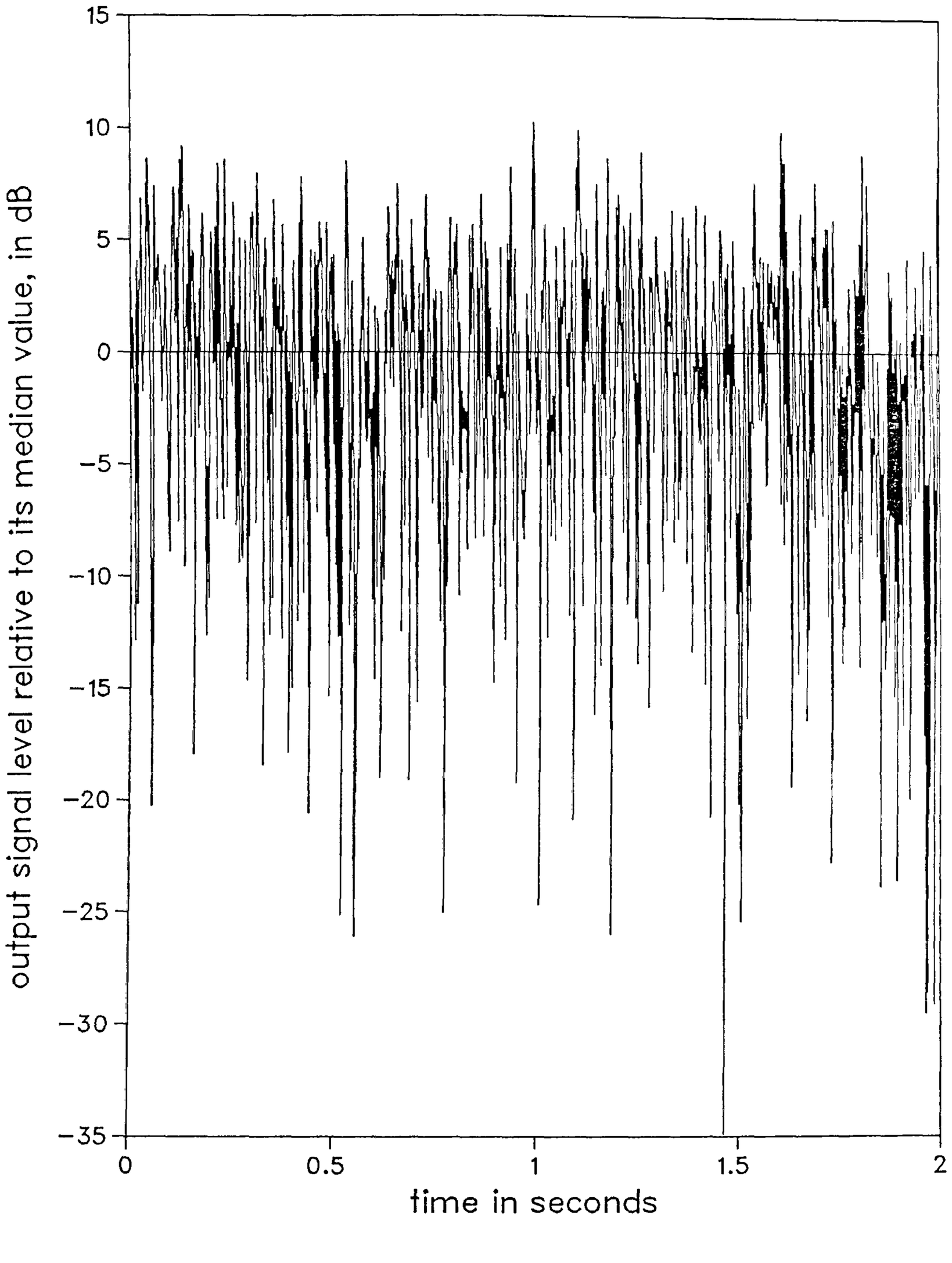


Fig.6.6.1. Signal–level fading in transmission path TX0

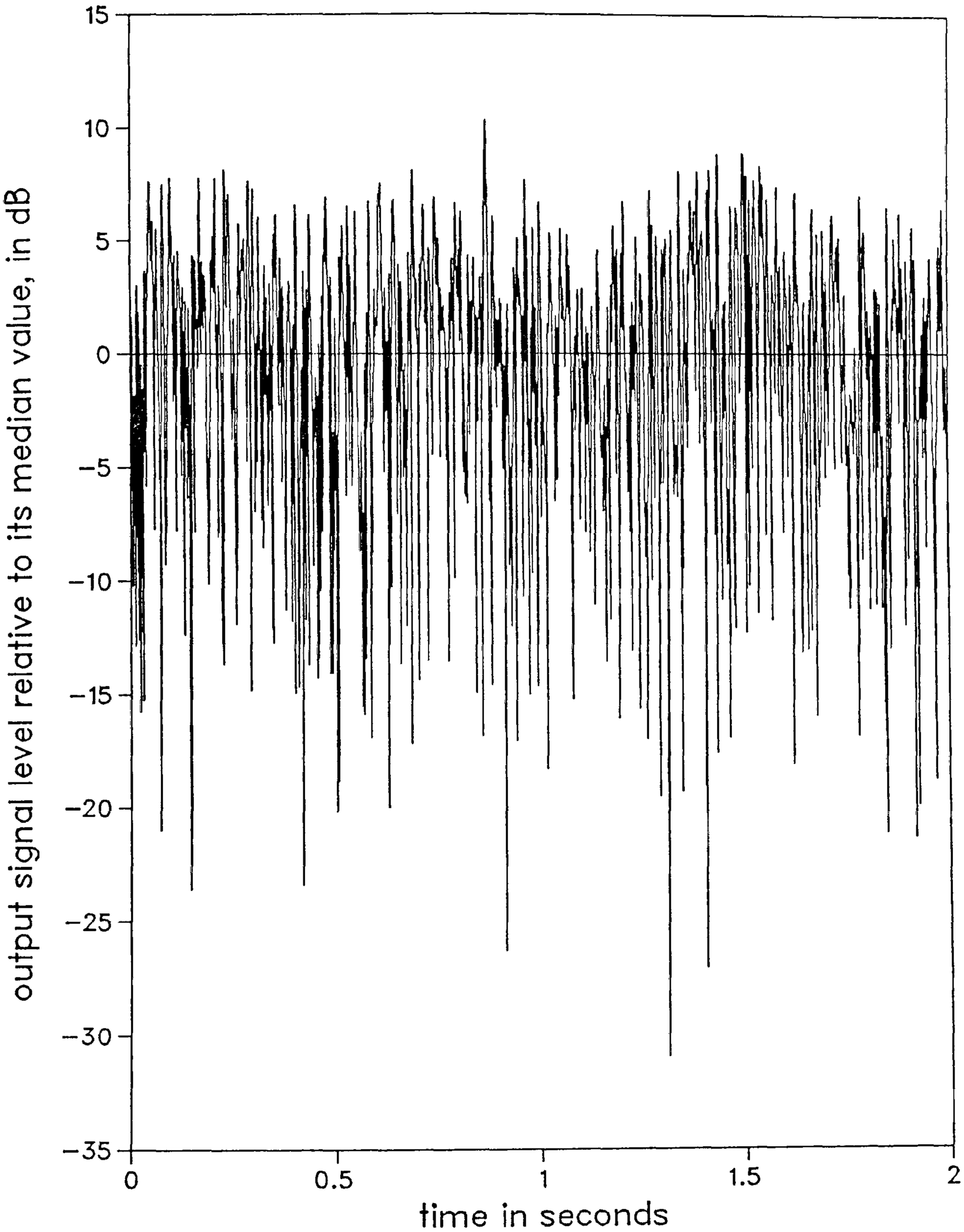


Fig.6.6.2. Signal–level fading in transmission path TX1

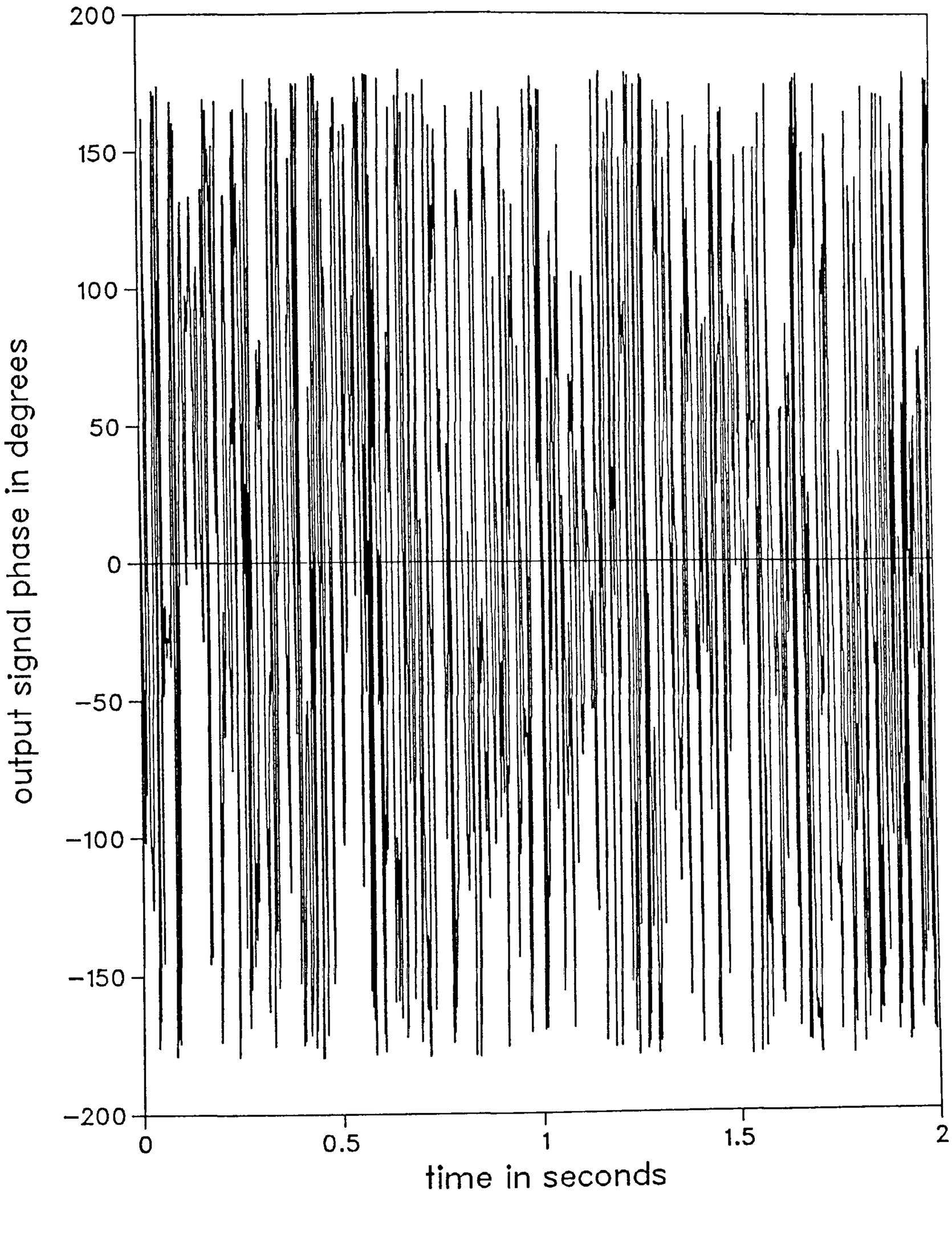


Fig.6.6.3. Phase variations in transmission path TX0

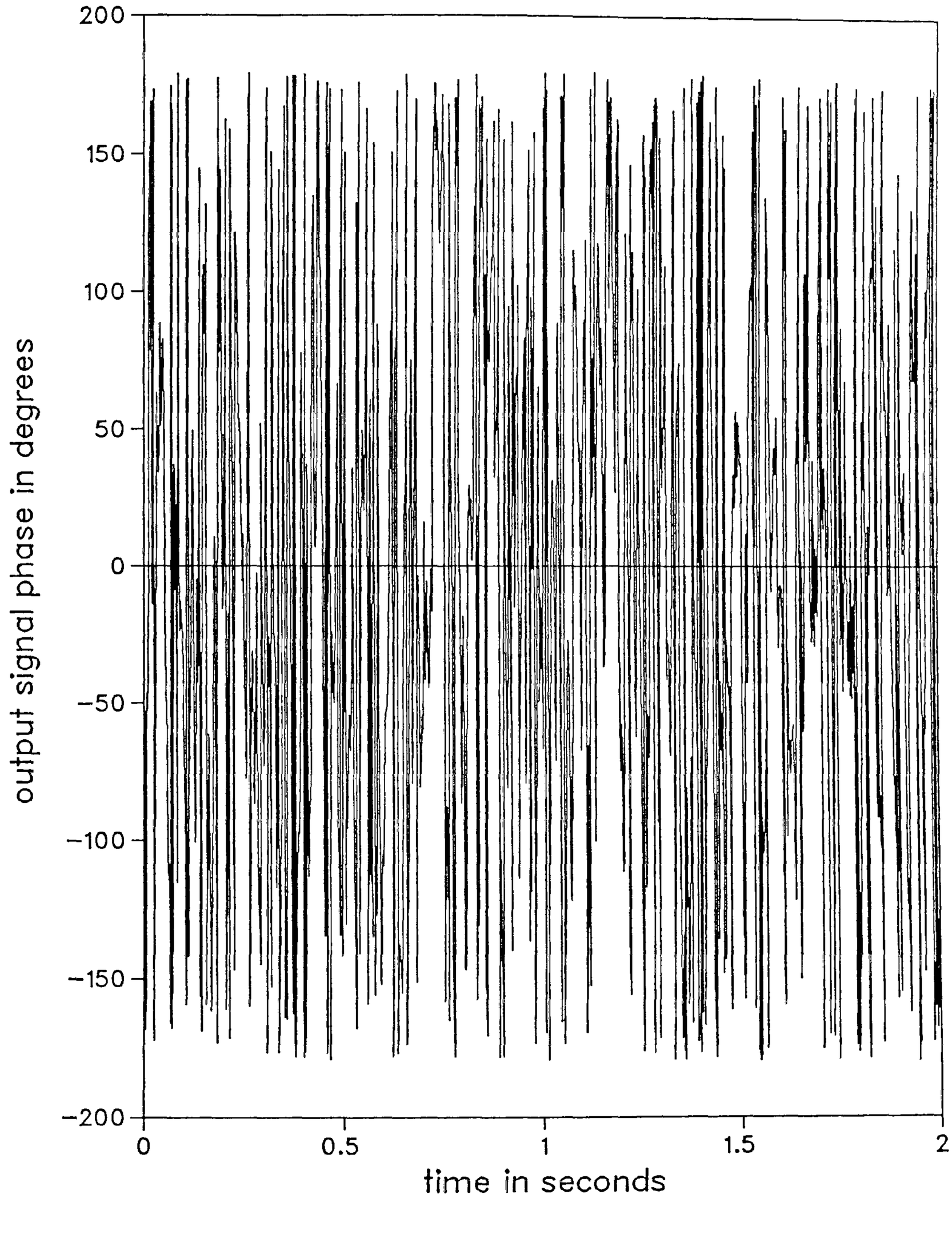


Fig.6.6.4. Phase variations in transmission path TX1

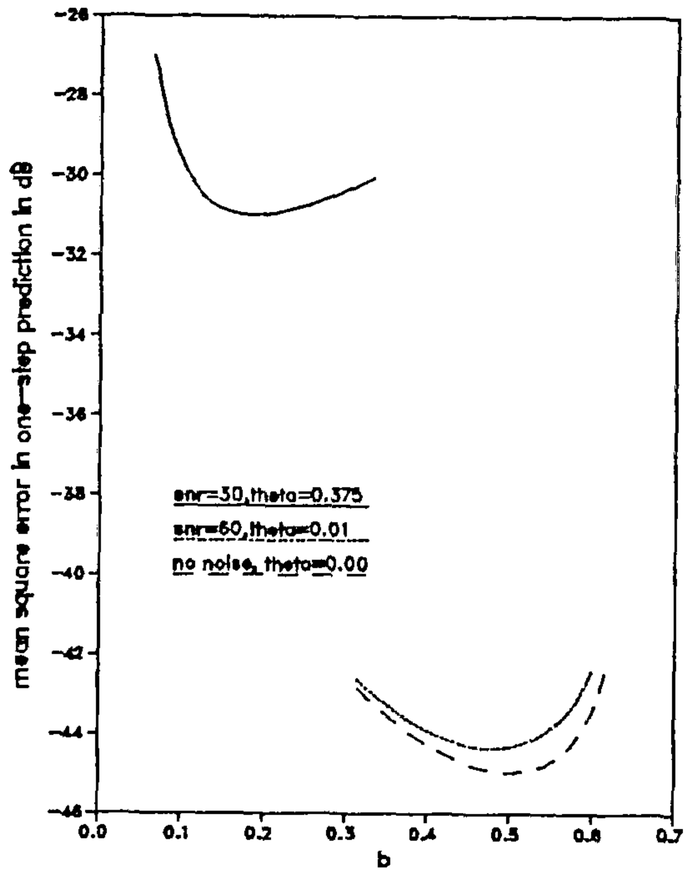


Fig. 6.6.5. Performance of MGE with degree-1 LSFM prediction, for one signal

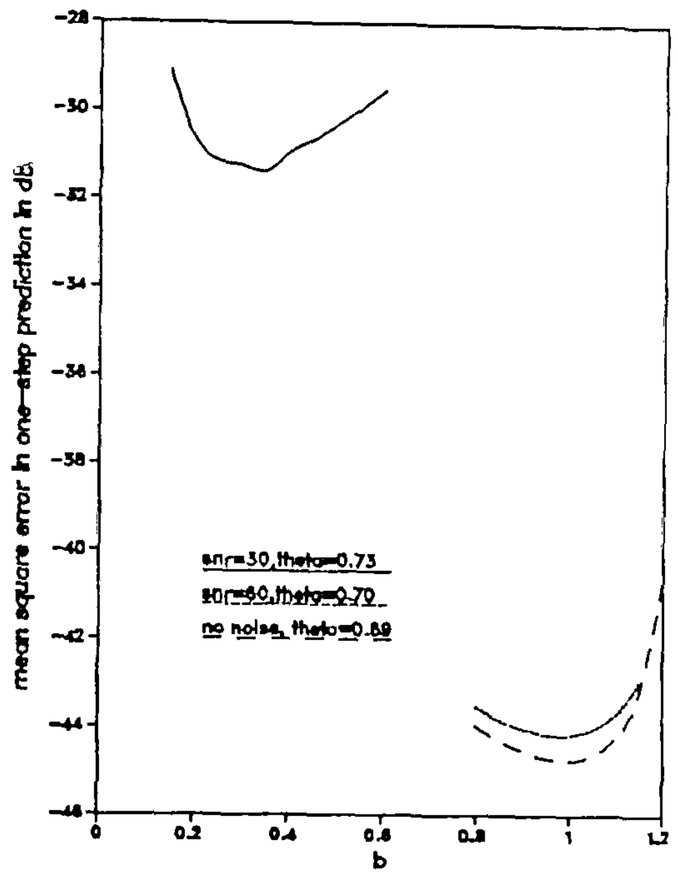


Fig. 6.6.6. Performance of MGE with degree-2 LSFM prediction, for one signal

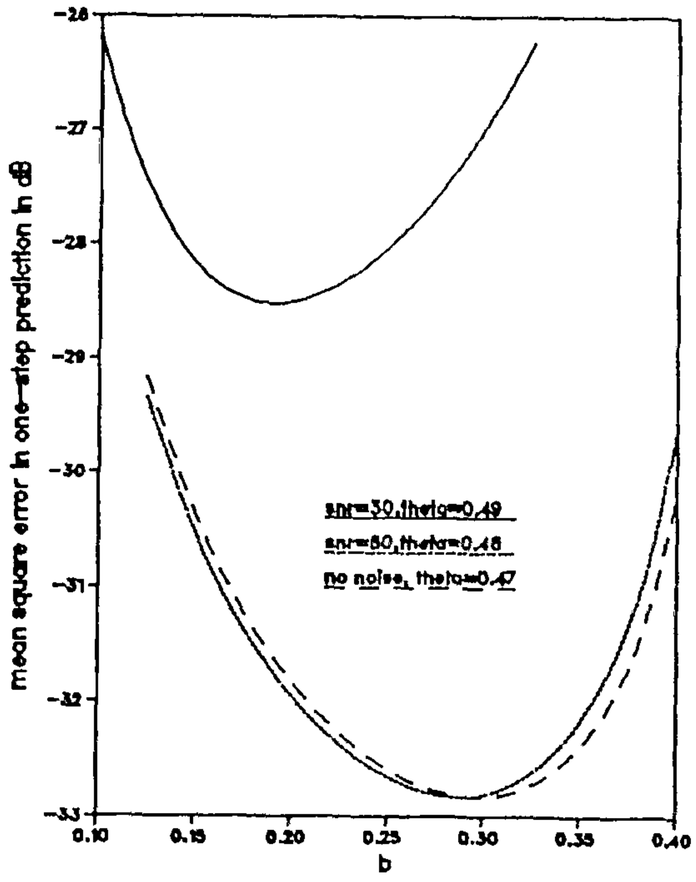


Fig. 6.6.7. Performance of MGE with degree-1 LSFM prediction, for two signals

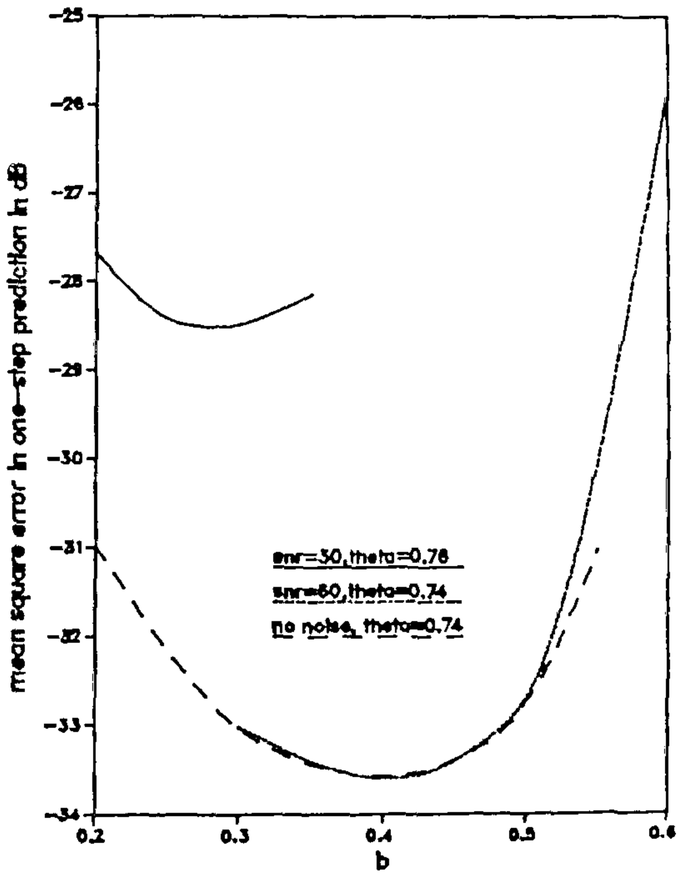


Fig. 6.6.8. Performance of MGE with degree-2 LSFM prediction, for two signals

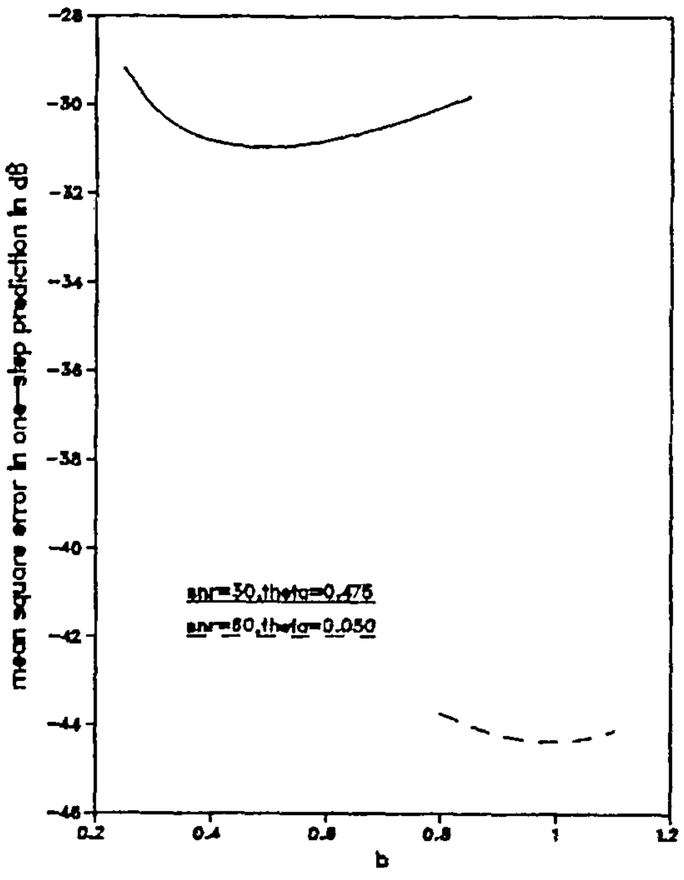


Fig. 6.6.9. Performance of UE with degree-1 LSFM prediction, for one signal

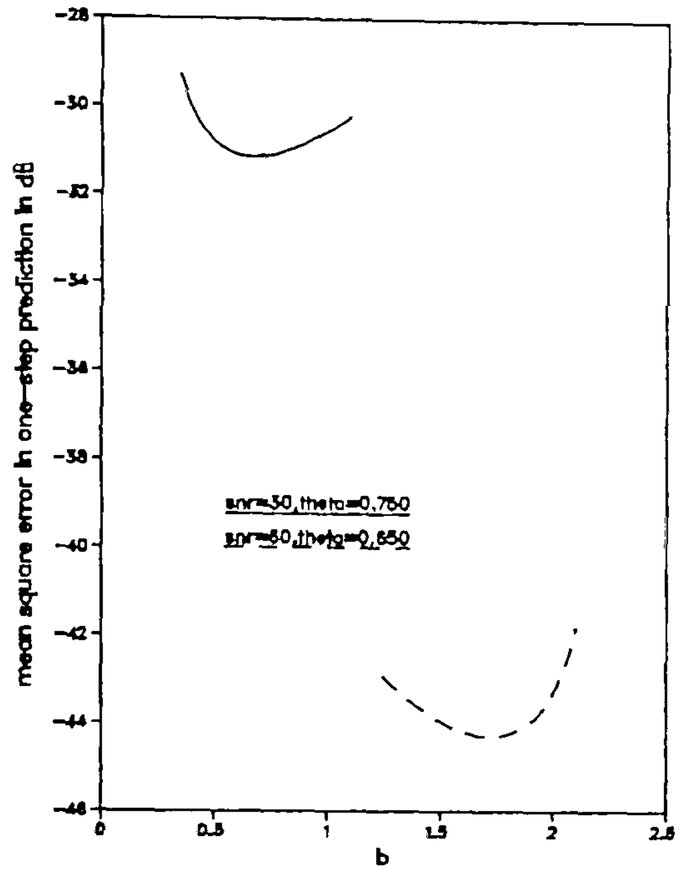


Fig. 6.6.10. Performance of UE with degree-2 LSFM prediction, for one signal

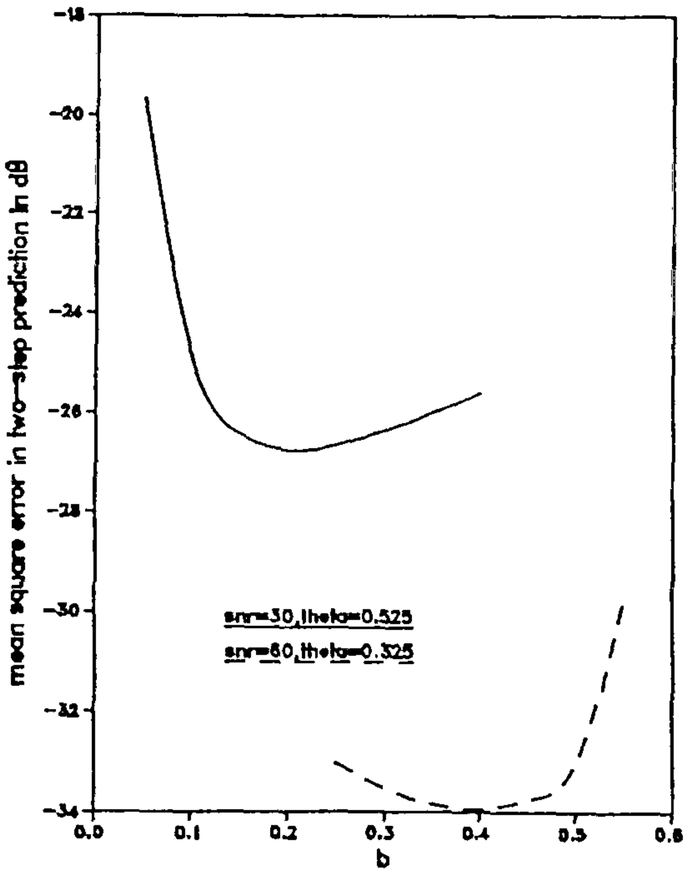


Fig. 6.6.11. Performance of UE with degree-1 LSFM prediction, for two signals

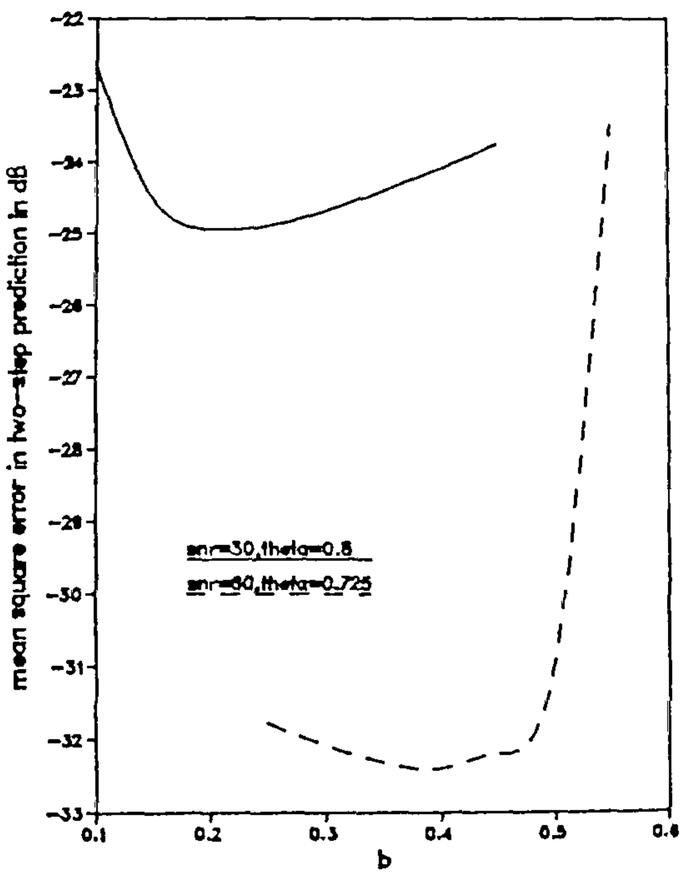


Fig. 6.6.12. Performance of UE with degree-2 LSFM prediction, for two signals

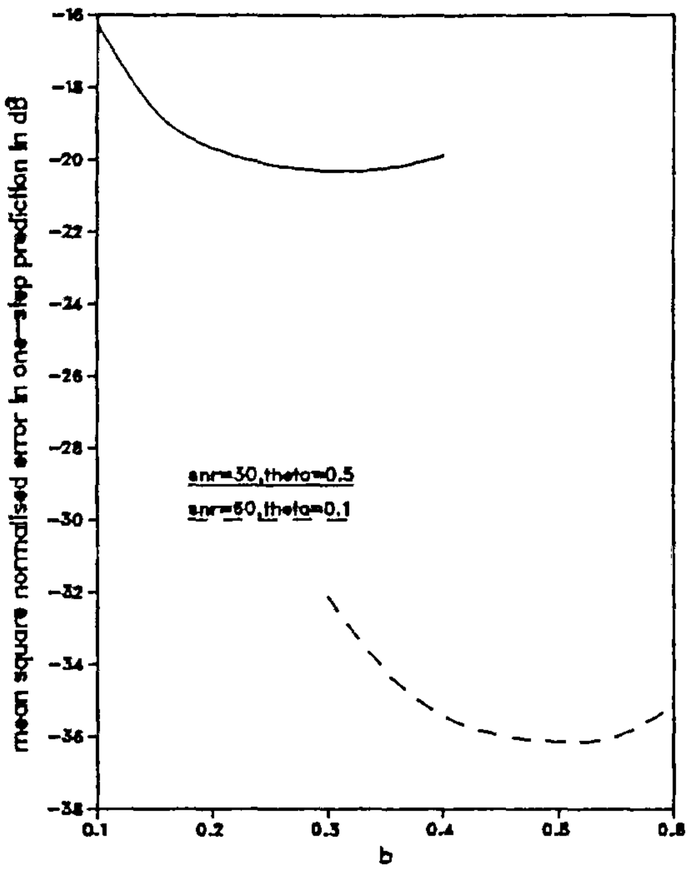


Fig. 6.6.13. Performance of MGE with degree–1 LSFM prediction, for one signal

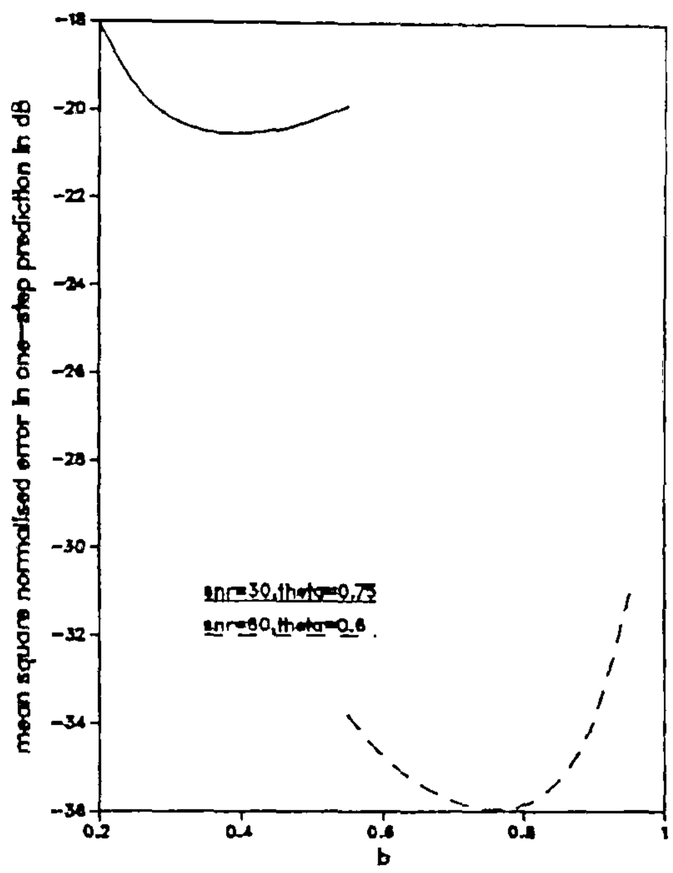


Fig. 6.6.14. Performance of MGE with degree–2 LSFM prediction, for one signal

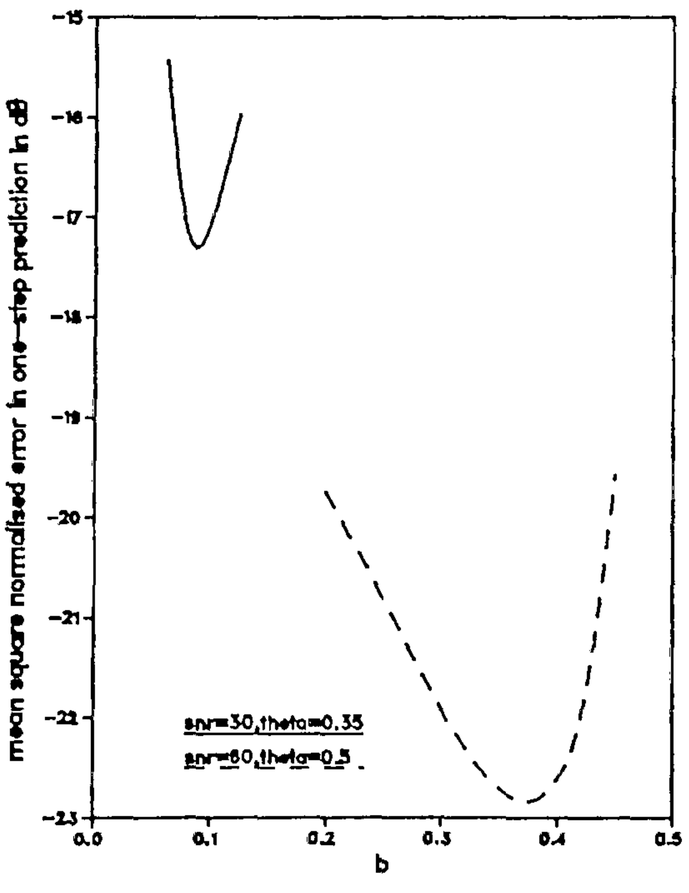


Fig. 6.6.15. Performance of MGE with degree–1 LSFM prediction, for two signals

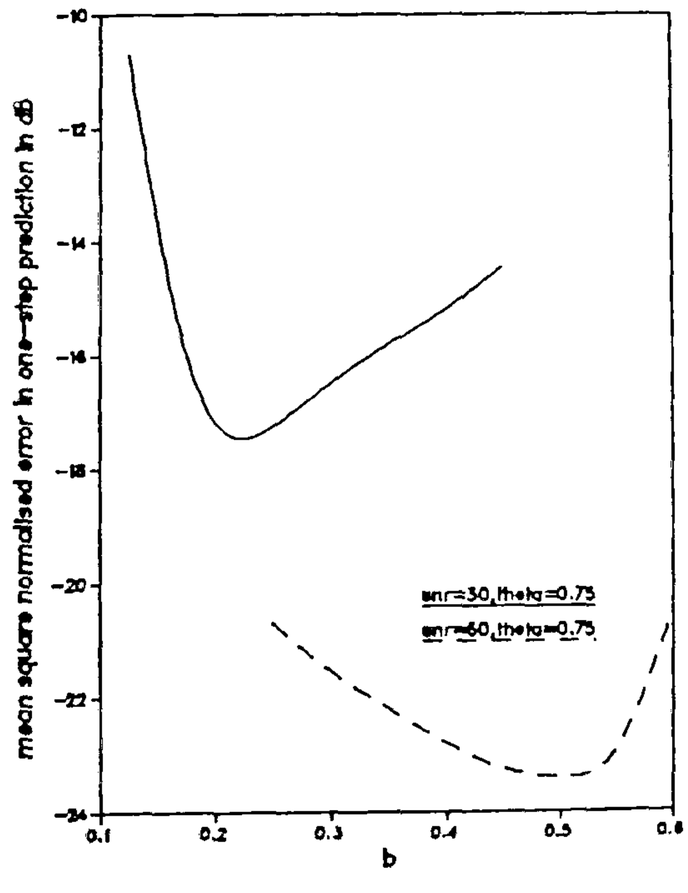


Fig. 6.6.16. Performance of MGE with degree–2 LSFM prediction, for two signals

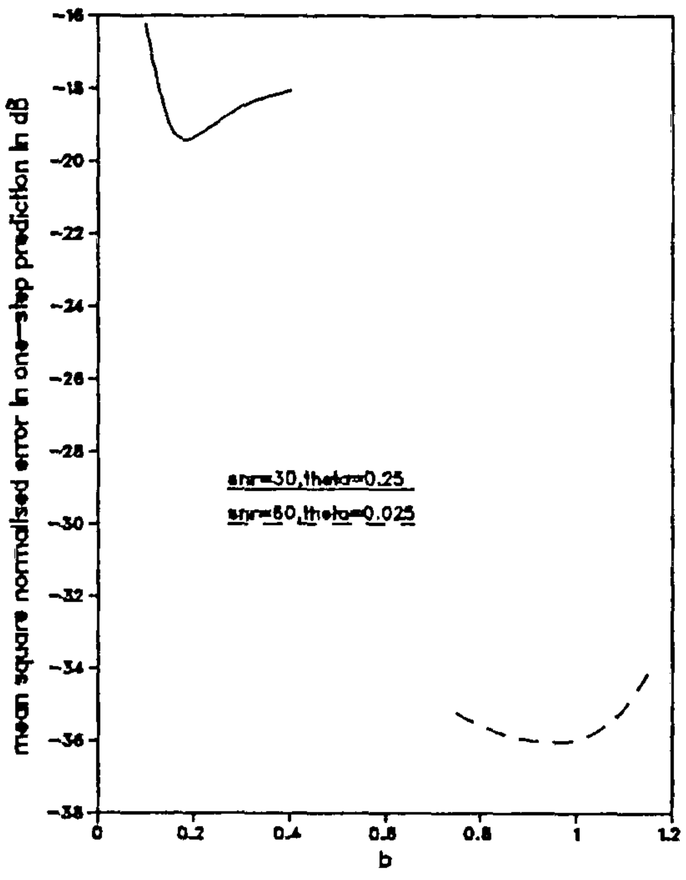


Fig. 6.6.17. Performance of UE with degree-1 LSFM prediction, for one signal

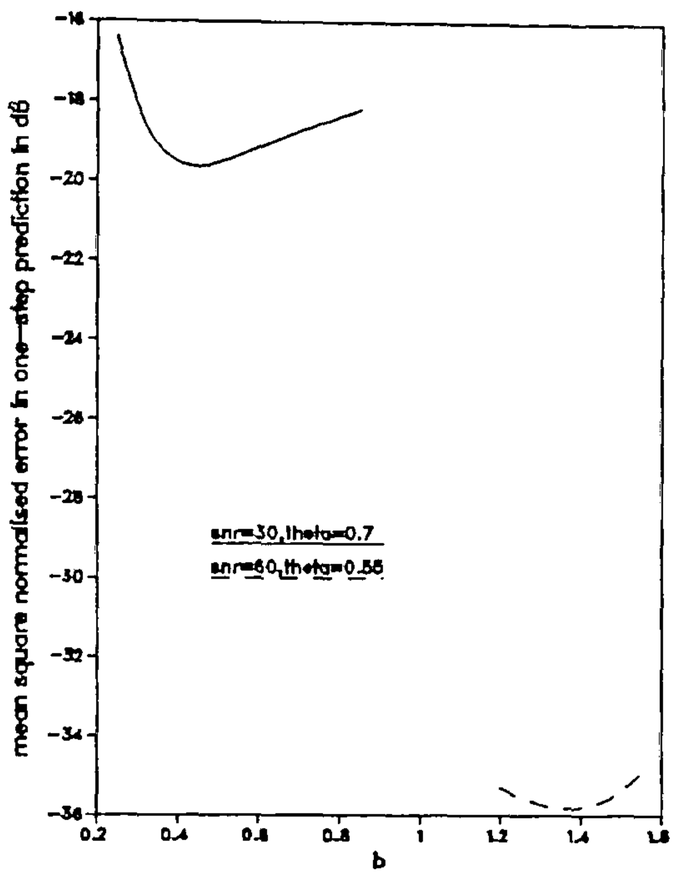


Fig. 6.6.18. Performance of UE with degree-2 LSFM prediction, for one signal

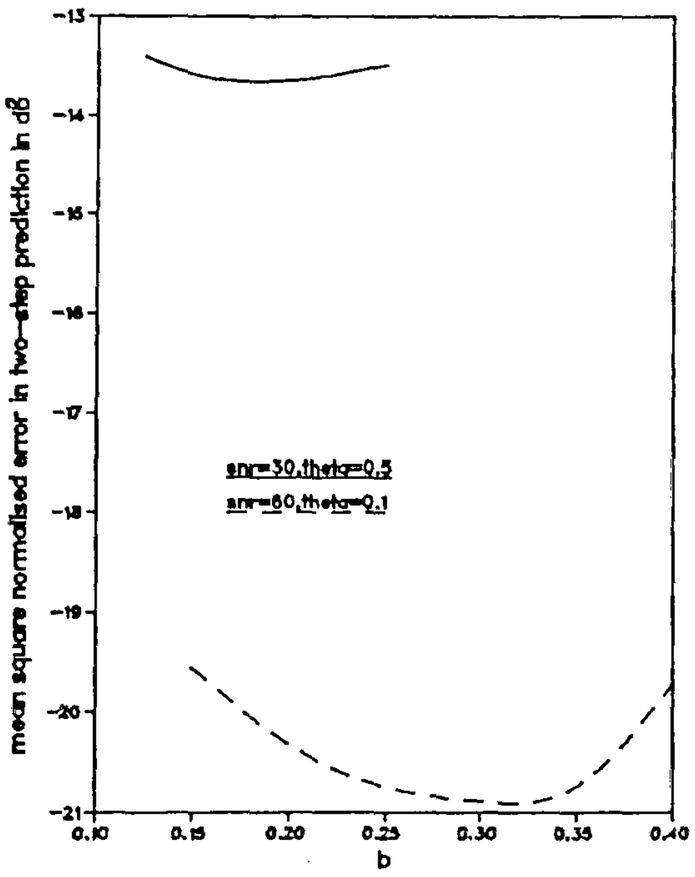


Fig. 6.6.19. Performance of UE with degree-1 LSFM prediction, for two signals

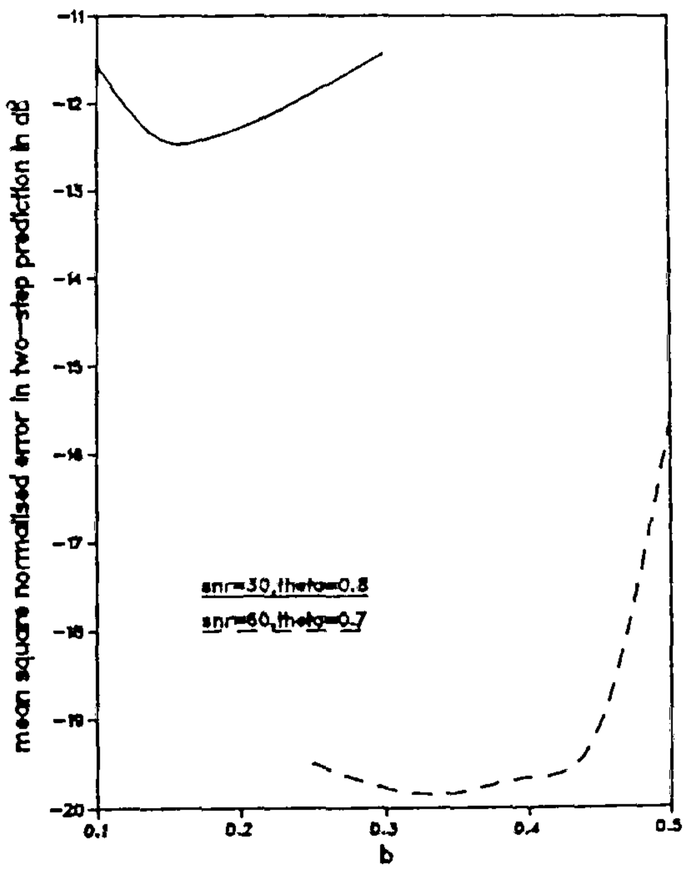


Fig. 6.6.20. Performance of UE with degree-2 LSFM prediction, for two signals

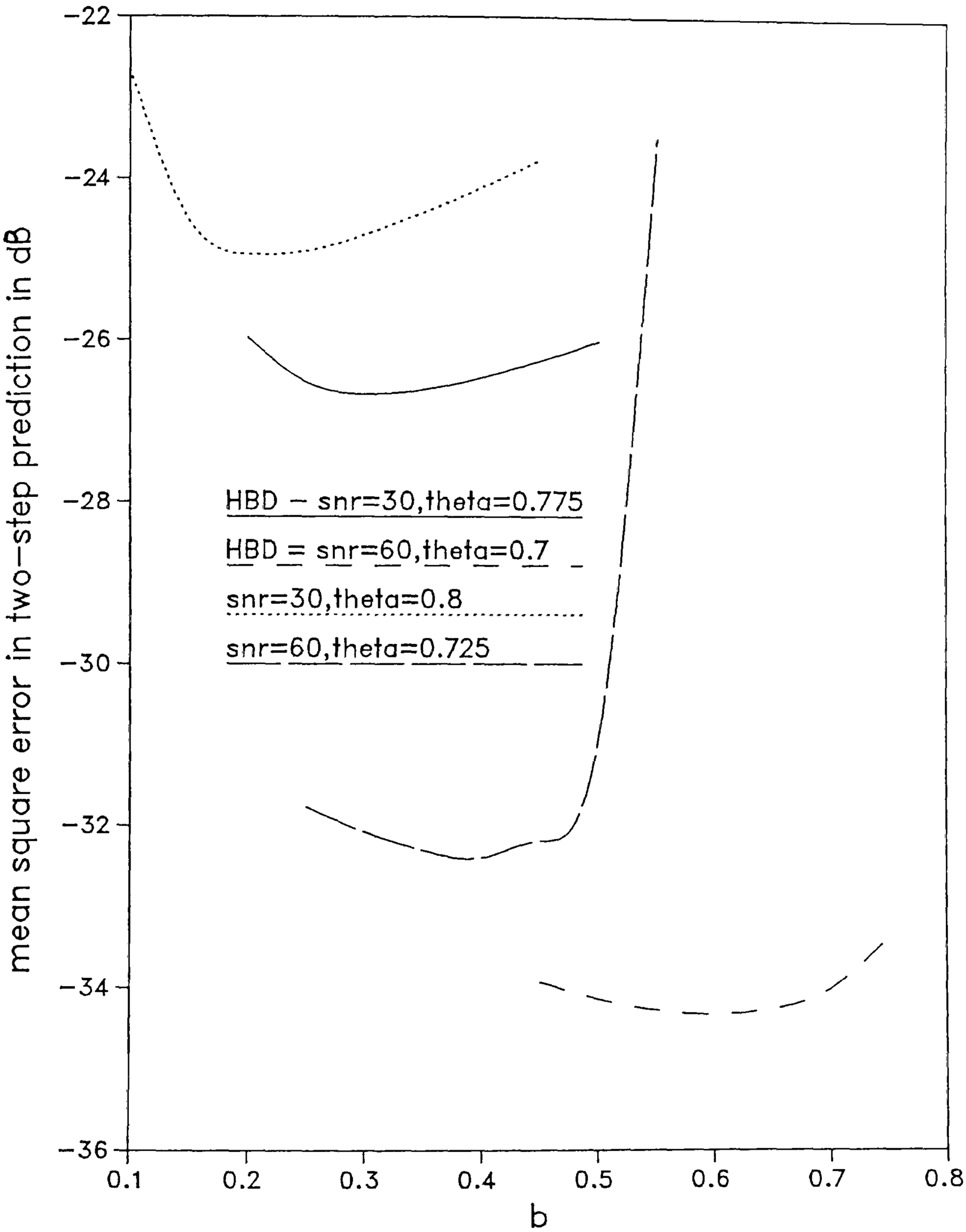


Fig. 6.6.21. Performance of hybrid (HBD) arrangement of UE with degree−2 LSFM prediction, for two signals

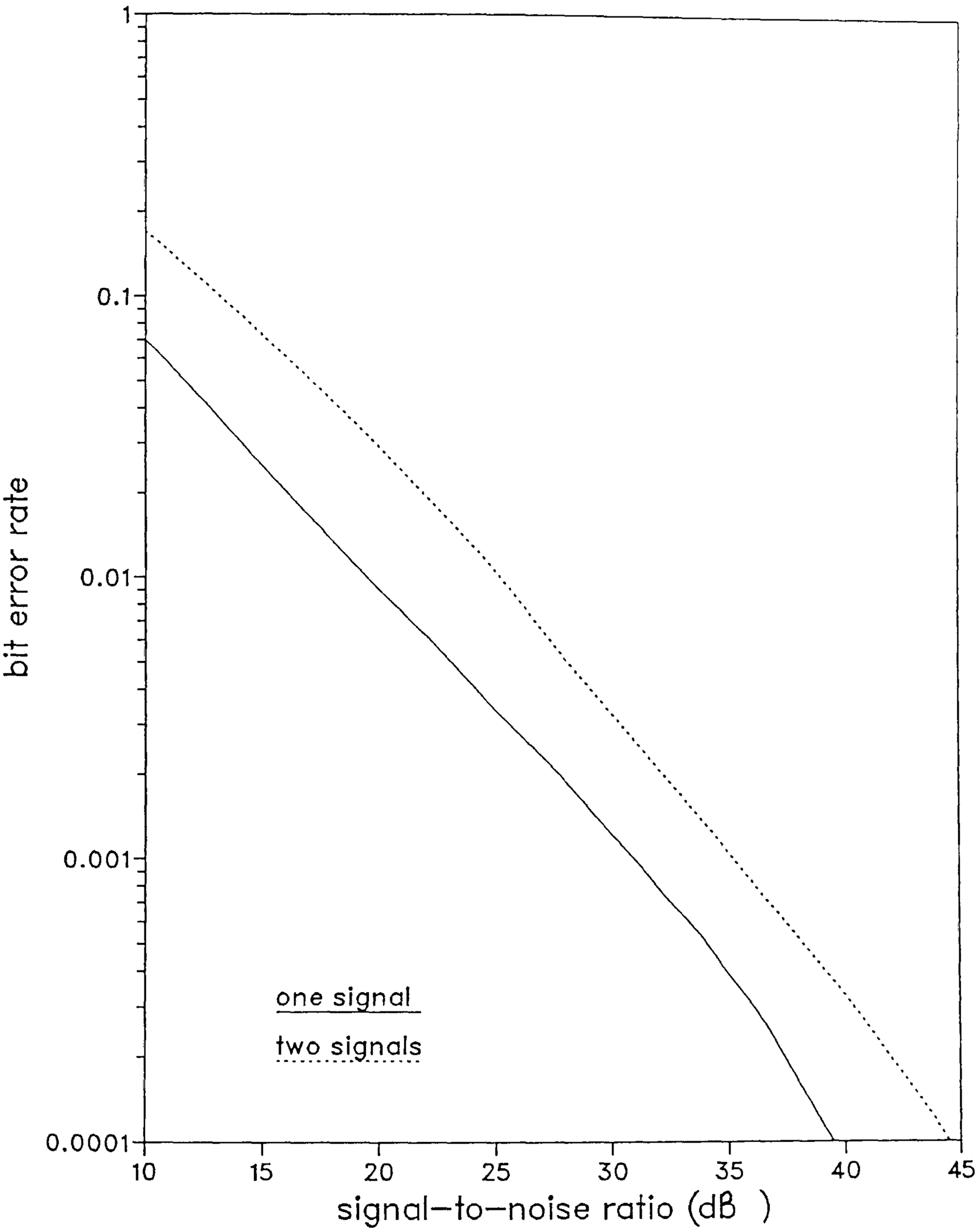


Fig.6.6.22. Performance of SD with perfect estimation

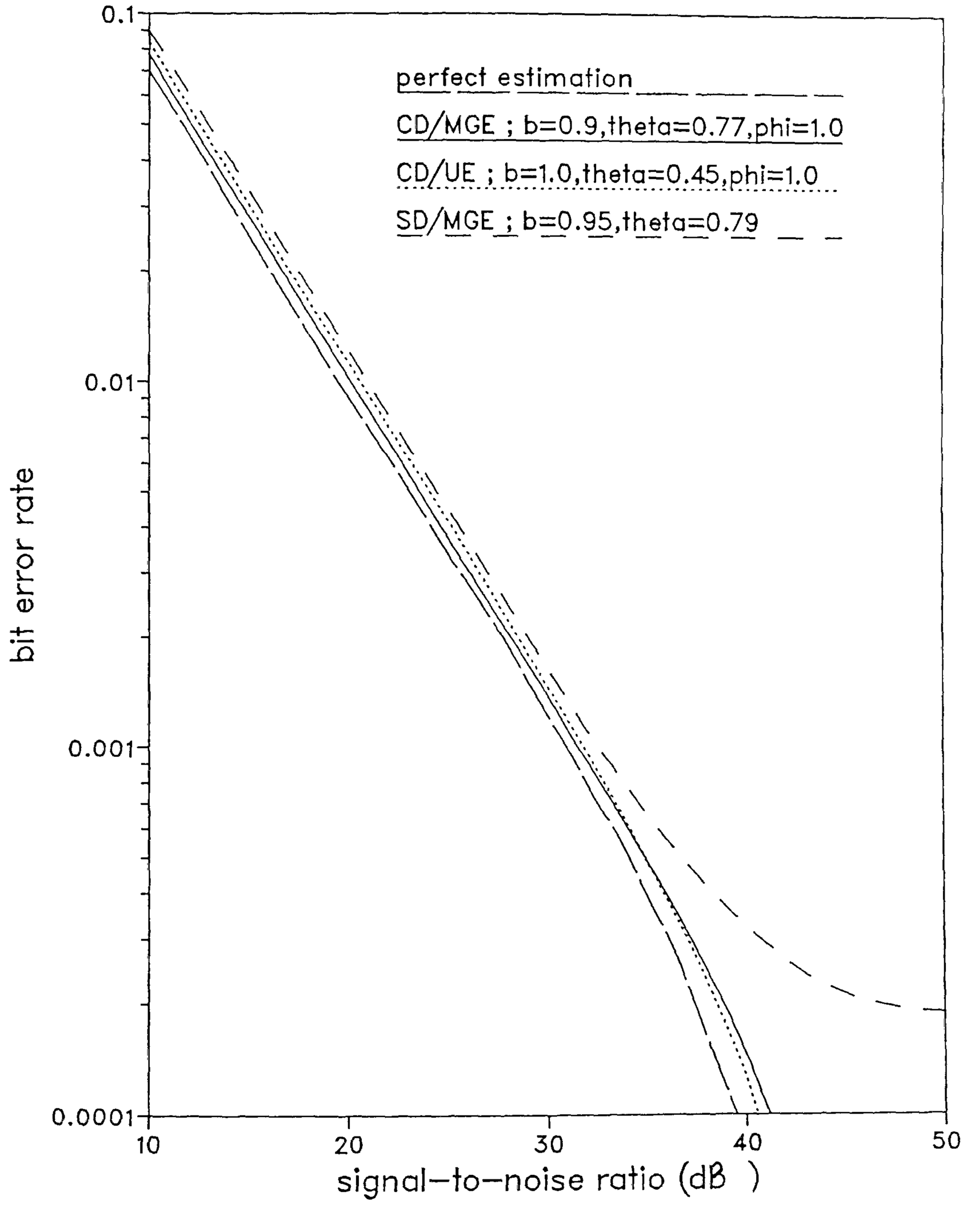


Fig.6.6.23. Performance of system with one signal using degree-1 LSFM prediction

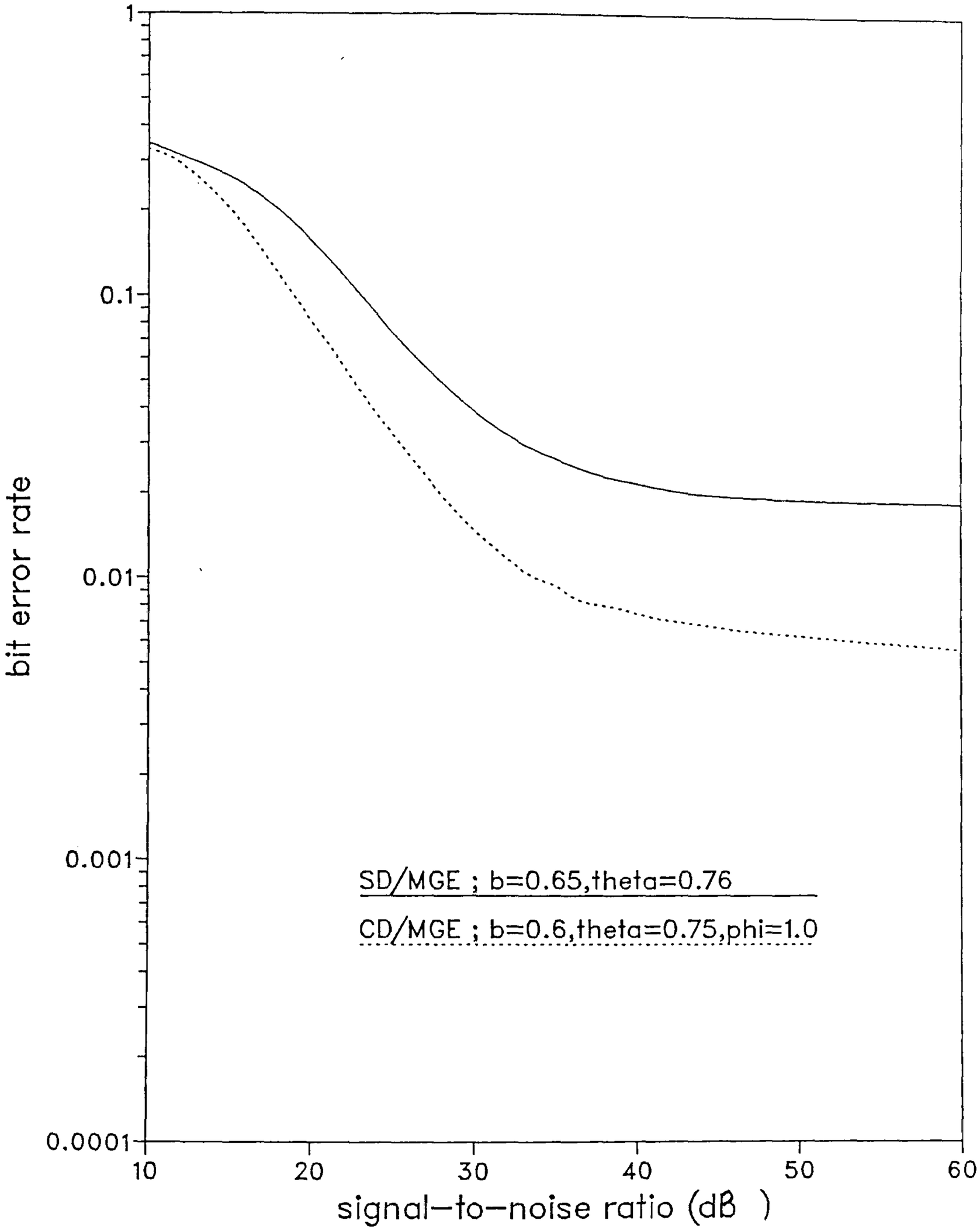


Fig.6.6.24. Performance of system with two signals using degree–1 LSFM prediction and retraining of estimates

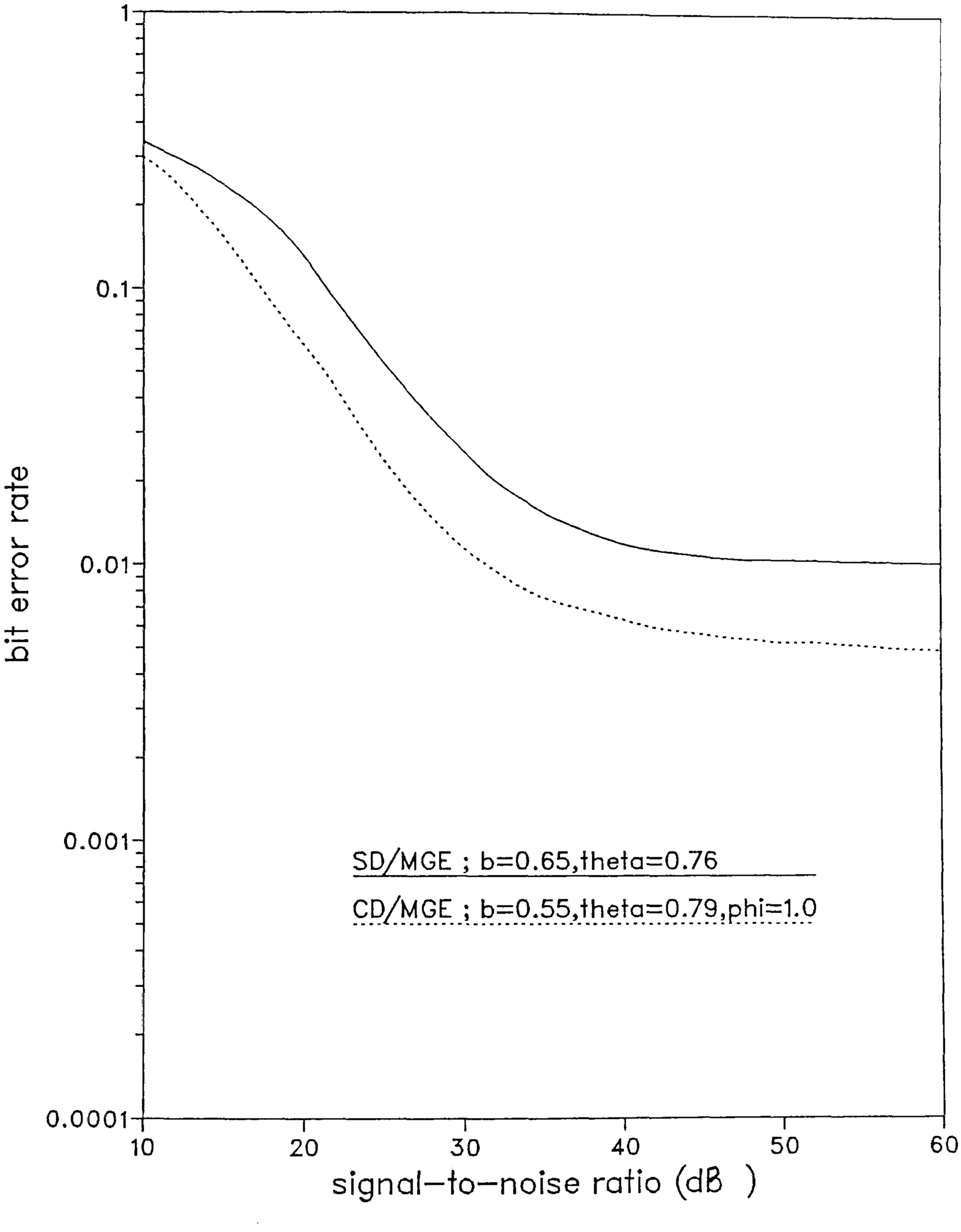


Fig.6.6.25. Performance of system with two signals using degree–1 LSFM prediction and retraining of estimates and their rates of change

CHAPTER 7

COMBINED DETECTION AND ESTIMATION FOR A 4.8 KBIT/S HF RADIO MODEM

7.1 INTRODUCTION

The detectors developed in Chapter 4 were tested under the assumption that the receiver has exact prior knowledge of the channel, i.e., perfect estimation was assumed. Although a logical (and indeed, necessary) assumption for testing the relative performance of various detectors, the corresponding results may not give an accurate indication of the performance of the system when the receiver has to estimate the sampled impulse response of the channel, as would be the case in practice. Furthermore, the minimum phasing of the channel was carried out in Chapter 4, with the aid of a root-finding software module from the Numerical Algorithm Group (NAG) library, along with some attendant calculations. Again, this is not a practically feasible method for generating a minimum phase channel. In practice, a minimum phase channel would be generated by some sort of linear filter [1]. Thus, it is of utmost interest to test the performance of the system when the rather idealized assumptions mentioned above are not made.

This chapter describes the performance of the 4.8 kbit/s HF radio modem (Chapter 4) when the detector operates in conjunction with an estimator. The estimator used here is the appropriately configured modified gradient estimator described in Chapter 6. Moreover, a minimum phase channel is achieved here by employing an adaptive linear feedforward transversal filter just ahead of the detector, thus making the whole system practically feasible. Trellis coded modulation however, is not included here, largely due to the rather disappointing performance exhibited by 8-PSK coded modulation systems when operating over the given HF radio channels (Chapter 5).

Before summarizing the structure of the chapter, it is necessary first to introduce to the reader in slightly more detail, the adaptive linear feedforward transversal filter that is used just ahead of the detector. The particular filter used here was originally developed for telephone modems [2]. In its ideal form, the filter is an allpass network with an infinite number of taps, and it adjusts the sampled impulse response of the channel and filter to be a minimum phase. The filter achieves this without changing any amplitude distortion in the channel and without changing the levels of the data signal and noise, thus avoiding any noise enhancement.

The adjustment of the filter is based on a novel technique that has been developed for telephone modems [2]. In addition to this adjustment, the algorithm also gives an estimate of the sampled impulse response of the channel and filter, which is ideally, minimum phase. The sole input required by the filter is an estimate of the sampled impulse response of the channel, which is provided by the channel estimator. Given this estimate of the sampled impulse response of the channel, the time taken for the adjustment of the filter appears to be much less than that needed to obtain the said initial estimate [2] . Thus, the total time taken for adjustment of the system is not much greater than that needed to estimate the channel. Now, a channel estimate could be obtained much more quickly and accurately than the corresponding adjustment of the adaptive filter by conventional means [2] and as such, the new technique has a much shorter convergence time than more conventional systems [3,4] .

The chapter continues with a description in Section 7.2, of the model of the data transmission system that was used in the tests. Section 7.3 describes the detector and Section 7.4 describes the estimator used, that is, the modified gradient estimator that was developed in Chapter 6, but in this case, configured to operate over a channel with memory. Section 7.5 gives a rather qualitative description of the adaptive linear feedforward transversal filter that is used ahead of the detector. The algorithm for the adjustment of the filter has been described in detail elsewhere [2], and hence is not included in the main body of the thesis. However, for the sake of completeness, the adjustment algorithm is given, along with the appropriate modifications for operation over time varying channels, in Appendix F. The chapter concludes in Section 7.6, with a discussion of the computer simulation tests and their results.

7.2 MODEL OF DATA TRANSMISSION SYSTEM

The model of the system that is used in the tests, is given in Fig. 7.2.1. The initial section of this model, upto the sampler, is exactly the same as that described in Section 4.4 and given in Fig. 4.4.1, with just one exception. The equivalent time varying linear baseband channel is not now, a minimum phased version. Rather, it is the 'raw' channel whose response is given by $y''_i(t\text{-}iT)$ in eqn. 4.4.7. In the interest of clarity however, this response will be denoted in the sequel, as $y_i(t\text{-}iT)$ (which was the notation for the scaled minimum phased channel in Chapter 4 - eqn. 4.4.14). Thus, the reader is reminded again, that $y_i(t\text{-}iT)$ as used in this chapter, does not denote the minimum phased channel, but is the original channel (eqn. 4.4.7) from which the minimum phased channel was derived in Chapter 4. Since the rest of the model, up to the sampler in Fig. 7.2.1, is

exactly the same as that in Fig. 4.4.1, its description is not repeated here.

Consider Fig. 7.2.1. The received, baseband signal waveform r(t), is sampled once per data symbol, and the output of the sampler at t=iT is a complex valued sample given by,

$$r_i = \sum_{h=0}^{g} s_{i-h} y_{i,h} + w_i \qquad (7.2.1)$$

where,

$$y_{i,h} = y_{i-h}(hT) \qquad (7.2.2)$$

and

$$w_i = w(iT) \qquad (7.2.3)$$

It is assumed that $y_{i,h} = 0$ for $h < 0$ and $h > g$ and the vector

$$Y_i = [\, y_{i,0} \quad y_{i,1} \quad \ldots\ldots \quad y_{i,g} \,] \qquad (7.2.4)$$

with z-transform

$$Y_i(z) = y_{i,0} + y_{i,1} z^{-1} + \ldots\ldots + y_{i,g} z^{-g} \qquad (7.2.5)$$

is taken to be the complex valued 'sampled impulse response' of the equivalent linear baseband channel at t = iT. The $\{w_i\}$ are samples of the stationary, zero mean, complex valued noise waveform w(t). As shown in Section 4.4, the real and imaginary parts of the $\{w_i\}$ are Gaussian random variables with zero mean and variance σ^2. The neighbouring $\{w_i\}$ are slightly correlated, and the reasons for, and consequences of this correlation was fully described in Section 4.4.

The received sample r_i, is now passed to the adaptive filter, which is a linear feedforward transversal filter with (n+1) taps, where the tap gains are given by

$$D_i = [\, d_{i,0} \quad d_{i,1} \quad \ldots\ldots \quad d_{i,n} \,] \tag{7.2.6}$$

with z-transform

$$D_i(z) = d_{i,0} + d_{i,1} z^{-1} + \ldots\ldots + d_{i,n} z^{-n} \tag{7.2.7}$$

The output signal from this filter at t = iT is

$$v_i = \sum_{h=0}^{n} r_{i-h}\, d_{i,h} \tag{7.2.8}$$

From eqns. 7.2.1 and 7.2.8, v_i could be represented as

$$v_i = \sum_{h=0}^{n+g} s_{i-h}\, e_{i,h} + w_i' \tag{7.2.9}$$

where the sequence

$$E_i = [\, e_{i,0} \quad e_{i,1} \quad \ldots\ldots \quad e_{i,n+g} \,] \tag{7.2.10}$$

has the z-transform

$$E_i(z) = e_{i,0} + e_{i,1} z^{-1} + \ldots\ldots + e_{i,n+g} z^{-(n+g)} \tag{7.2.11}$$

and w_i' is a noise sample originating from w_i .

The filter in fact, is such that $e_{i,h} \approx 0$ for h = 0, 1, ..., (n-1), and so introduces a delay of n sampling intervals such that the earliest symbol that could be detected at t = iT is s_{i-n} . In addition to v_i , the filter gives at its output, the sequence

$$F_i = [\, f_{i,0} \quad f_{i,1} \quad \ldots\ldots \quad f_{i,g} \,] \tag{7.2.12}$$

with z-transform

$$F_i(z) = f_{i,0} + f_{i,1} z^{-1} + ... + f_{i,g} z^{-g} \qquad (7.2.13)$$

where $f_{i,h} \approx e_{i,n+h}$, for $h = 0, 1, ..., g,$ and F_i is an estimate of the value of the sampled impulse response of the cascade of the linear baseband channel at $t = (i-n)T$, and the adaptive filter. When the filter is correctly adjusted, $f_{i,h} = e_{i,n+h}$ for $h = 0,1, ..., g,$ and the sequence F_i should be minimum phase, i.e., $F_i(z)$ should have no roots outside the unit circle. In determining F_i the sole input used by the adaptive filter is an estimate (to be more precise, a prediction) of the sampled impulse response of the linear baseband channel at $t = (i-n)T$. This prediction is denoted $Y'_{i-n,i-n-1}$, where,

$$Y'_{i-n,i-n-1} = [\, y'_{i-n,i-n-1,0} \quad y'_{i-n,i-n-1,1} \quad ..... \quad y'_{i-n,i-n-1,g} \,] \qquad (7.2.14)$$

and is that which was the output of the channel estimator at $t = (i-1)T$.

The detector uses v_i and F_i to determine the detected value s'_{i-n-l}, of the data symbol s_{i-n-l}. Clearly, the detector itself introduces a delay of l sampling intervals so that the total delay in detection is $(n+l)$ sampling intervals. The detected symbol value s'_{i-n-l} is finally fed to a differential decoder, which performs an inverse operation to that of the differential encoder (Appendix B), to give at its output the detected binary digit α'_{i-n-l}.

In addition to s'_{i-n-l}, the detector also provides the 'early' detected data symbols $\{s''_{i-n-h}\}$, ($h = 0, 1, ..., g,$ and $g < l$), for use in the channel estimator. The channel estimator uses the appropriate received samples, together with the $\{s''_{i-n-h}\}$, to predict the value of the sampled impulse response of the linear baseband channel at $t = (i-n+1)\,T$ i.e., $Y'_{i-n+1,i-n}$, which will be required by the adaptive filter at the next sampling instant, $t = (i+1)T$.

The model given above assumes ideal symbol timing. The following sections explain the operation of the detector, estimator and adaptive linear feedforward filter in greater depth. For the sake of convenience, the adaptive linear feedforward filter will henceforth be referred to as 'the adaptive filter', and the equivalent linear baseband channel will be referred to simply as 'the channel'.

7.3 THE DETECTOR

The detector used in the tests is System C (Section 4.5.3) with eight stored vectors, i.e., C8. This is the detector that was chosen from earlier tests (see Chapter 4), as that giving the best compromise between performance and complexity. The operation of the detector is exactly as described in Section 4.5.3, except that it now operates on the sample v_i at the output of the adaptive filter, rather than on the received sample r_i. The other essential difference is that the knowledge it has of the channel is simply the vector F_i, which is an estimate of the cascade of the sampled impulse response of the channel at t = (i-n)T, and the adaptive filter.

Just prior to the receipt of the sample v_i, the detector holds in store eight different l-component vectors (sequences) $\{Q_{i-n-1}\}$ where

$$Q_{i-n-1} = [x_{i-n-l} \quad x_{i-n-l+1} \quad \ldots\ldots \quad x_{i-n-1}] \tag{7.3.1}$$

with x_i taking on any of the possible values of s_i. Each of these vectors has associated with it a 'cost', given by,

$$|U_{i-n-1}|^2 = \sum_{j=n}^{i-1} \Big| v_j - \sum_{h=0}^{g} x_{j-n-h} f_{j,h} \Big|^2 \tag{7.3.2}$$

where |x| is the absolute value of the scalar quantity x, and $x_i = 0$ for $i \leq 0$.

On receipt of the sample v_i, the vectors $\{Q_{i-n-1}\}$ are expanded as per System C (Section 4.5.3), to give the $(l + 1)$-component vectors $\{P_i\}$, where,

$$P_i = [x_{i-n-l} \quad x_{i-n-l+1} \quad \ldots\ldots \quad x_{i-n-1} \quad x_{i-n}] \tag{7.3.3}$$

Stored alongside each vector P_i, is its cost which is now given by

$$|U_{i-n}|^2 = |U_{i-n-1}|^2 + \Big| v_i - \sum_{h=0}^{g} x_{i-n-h} f_{i,h} \Big|^2 \tag{7.3.4}$$

The expansion of the vectors, selection of the best vector and subsequent retention of vectors for the next time instant t = (i+1)T, is exactly as described in Section 4.5.3, with the detector now providing, in addition to the final detected data symbol s'_{i-n-l}, the early detected data symbols $\{s''_{i-n-h}\}$, (h = 0, 1, ..., g), for use in the channel estimator.

7.4 THE ESTIMATOR

The estimator used in the tests is the modified gradient estimator described in Section 6.4.2, but configured now, to operate on channels with memory. Its operation is as follows.

At the time instant t = iT, the estimator receives the 'early' detected data symbols $\{s''_{i-n-h}\}$, (h = 0, 1, ... g), from the detector. In addition to this, it is also fed the received signal corresponding to the time instant t = (i-n)T, i.e., r_{i-n}, as shown in Fig. 7.2.1, where,

$$r_{i-n} = \sum_{h=0}^{g} s_{i-n-h}\, y_{i-n,h} + w_{i-n} \qquad (7.4.1)$$

The estimator first forms an estimate of r_{i-n}, given by

$$r'_{i-n} = \sum_{h=0}^{g} s''_{i-n-h}\, y'_{i-n,i-n-1,h} \qquad (7.4.2)$$

where

$$Y'_{i-n,i-n-1} = [\, y'_{i-n,i-n-1,0} \quad y'_{i-n,i-n-1,1} \quad ..... \quad y'_{i-n,i-n-1,g} \,] \qquad (7.4.3)$$

and $Y'_{i-n,i-n-1}$ is a prediction of the channel corresponding to the time instant t = (i-n)T, which was formed by the estimator at the time instant t = (i-1)T.

The estimator then calculates the error, e'_{i-n}, in the estimate of the received signal, as

$$e'_{i-n} = r_{i-n} - r'_{i-n} \tag{7.4.4}$$

The updated estimate vector Y'_{i-n} is now formed, where,

$$Y'_{i-n} = [\, y'_{i-n,0} \quad y'_{i-n,1} \quad \ldots\ldots \quad y'_{i-n,g} \,] \tag{7.4.5}$$

and where the $(h+1)^{th}$ component of this vector is

$$y'_{i-n,h} = y'_{i-n,i-n-1,h} + b\, e'_{i-n}\, (s''_{i-n-h})^* \tag{7.4.6}$$

b here, is a small positive quantity known as the 'step size' (Section 6.4.2) and $(s''_{i-n-h})^*$ denotes the complex conjugate of s''_{i-n-h}.

The estimator now obtains the error in the updated estimate vector, to give the error vector Γ_{i-n}, where,

$$\Gamma_{i-n} = [\, \varepsilon_{i-n,0} \quad \varepsilon_{i-n,1} \quad \ldots\ldots \quad \varepsilon_{i-n,g} \,] \tag{7.4.7}$$

and where the $(h+1)^{th}$ component of the vector Γ_{i-n} is given by

$$\varepsilon_{i-n,h} = y'_{i-n,h} - y'_{i-n,i-n-1,h}$$

$$= b\, e'_{i-n}\, (s''_{i-n-h})^* \tag{7.4.8}$$

The components of Γ_{i-n} are finally fed to the degree-1 least squares fading memory prediction algorithm [5] (given in Table 6.4.1, albeit with the subscripts reversed!!), to obtain the one-step prediction vector

$$Y'_{i-n+1,i-n} = [\, y'_{i-n+1,i-n,0} \quad y'_{i-n+1,i-n,1} \quad \ldots\ldots \quad y'_{i-n+1,i-n,g} \,] \tag{7.4.9}$$

The vector $Y'_{i-n+1,i-n}$ is a prediction of the sampled impulse response of the channel

corresponding to the time instant t = (i-n+1)T. This prediction is that required by the adaptive linear feedforward transversal filter at time t = (i+1)T, i.e., at the next time instant.

7.5 THE ADAPTIVE FILTER

The adaptive filter used just ahead of the detector (Fig. 7.2.1) is a linear feedforward transversal filter with (n + 1) taps. At the time instant t = iT and before the detection of s'_{i-n-l}, the adaptive filter receives from the estimator, an estimate (more precisely, a prediction) of the sampled impulse response of the channel corresponding to the time instant t = (i-n)T. This estimate (or prediction) is of course $Y'_{i-n,i-n-1}$, as given in eqn. 7.2.14. The adaptive filter uses this prediction to both adjust the filter tap gains, and determine F_i (eqn. 7.2.12), where F_i is an estimate of the sampled impulse response of the cascade of the channel corresponding to the time instant t = (i-n)T, and the filter itself. When the filter is correctly adjusted, the sequence F_i should be minimum phase, i.e., all the roots of $F_i(z)$ (eqn. 7.2.13) should lie inside the unit circle.

A discussion of the theory of the adaptive filter and the adjustment algorithm for its taps, are considered in great detail in reference [2], albeit for the case of a time invariant channel. For the sake of clarity, it has been decided not to include the corresponding tap adjustment algorithm for time varying channels in the main text of the thesis, and so, is given in Appendix F. Thus, the reader who is interested in the general theory of the adaptive filter is referred to reference [2], and for a concise account of just the tap adjustment algorithm for time varying channels, he should refer to Appendix F. The objective of the remainder of this section is to describe the filter in a qualitative manner, from the point of view of its operation in the given HF modem receiver.

Consider the situation at t = iT. Since the filter introduces a delay of n sampling intervals, the earliest possible detected symbol is s'_{i-n}. The detector thus requires, at t = iT, an estimate of the sampled impulse response of the cascade of the channel corresponding to the time instant t = (i-n)T, and the filter. This sampled impulse response is, of course, the sequence F_i (eqn. 7.2.12) with z-transform $F_i(z)$ (eqn. 7.2.13), where F_i as generated by the filter is, ideally, a minimum phase sequence. (In practice however, F_i would only be near minimum phase since the ideal form of the filter assumes an infinite number of taps). The filter generates F_i via the following sequence of operations.

1. Accepts as input, an estimate (prediction) of the sampled impulse response of the channel corresponding to t = (i-n)T. This estimate (prediction) is the sequence $Y'_{i-n,i-n-1}$ (eqn. 7.2.14) with z-transform $Y'_{i-n,i-n-1}(z)$.

2. Determines, with the aid of a root finding algorithm, the roots of $Y'_{i-n,i-n-1}(z)$ which lie outside the unit circle. The root finding algorithm here is an iterative process (see Appendix F) which converges rapidly to those roots that lie outside the unit circle. The process stops when it has found all (or at least most) of the roots.

3. Forms the roots of $F_i(z)$ by replacing the roots of $Y'_{i-n,i-n-1}(z)$ that lie outside the circle, by the complex conjugates of their reciprocals, all remaining roots being left unchanged.

4. Generates F_i using the knowledge of the roots of $F_i(z)$.

In addition to estimating F_i, the filter uses its knowledge of these new set of roots to also adjust its tap gains. Thus, a new set of tap gains is obtained at every sampling instant, and at t = iT, these tap gains are given by D_i (eqn. 7.2.6) with z-transform $D_i(z)$ (eqn. 7.2.7). As mentioned earlier, the technique used for the adjustment of the taps is given in Appendix F.

7.6 COMPUTER SIMULATION TESTS AND RESULTS

Computer simulation tests have been carried out to assess the performance of the systems when operating over Channels 1 and 2, whose parameters are given in Table 2.5.2. Thus, Channel 1 is a 3-skywave channel with a frequency spread of 2 Hz. and relative delays of 1.1 and 3.0 ms. between the first skywave and the others, while Channel 2 is a 2-skywave channel with a frequency spread of 2 Hz. and a relative delay of 3 ms. between the skywaves. The fading sequence employed for a given channel, is exactly the same as that used in the tests in Chapter 4, where the performance of various near maximum likelihood detectors were obtained under certain idealized assumptions. Thus, the fading sequences are those represented in Figs. 4.6.1 and 4.6.2, where Channel 1 is the milder of the two channels in terms of the depths of the fades (see Section 4.6).

In all tests without the adaptive filter, perfect estimation is assumed, and the sampled impulse response of the channel has been made a minimum phase by employing a root finding algorithm from the NAG software library. This algorithm operates on the

roots of the z-transform of the channel that have absolute values greater than d, by replacing these roots by the complex conjugates of their reciprocals. The minimum phase channel is then determined with the aid of the new set of roots and some attendant calculations.

Although practically infeasible, the above method has the ability to track every root that has an absolute value greater than d, unlike the adaptive filter which may miss some roots, especially when the signal is in a deep fade. It thus provides an effective 'upper bound' in terms of the performance of the detectors, under the given conditions. In all tests, the value of d has been set to 1.05, instead of unity. This is done in order to ensure a proper comparison of the performance of the detector with and without the adaptive filter when perfect estimation is assumed, since a value of d=1 would have forced the filter to be used in its ideal form, which, of course, would require an infinite number of taps! Clearly, the performance curve of the detector without the adaptive filter is the same as that given in Figs. 4.6.8 and 4.6.12 for detector C8 over the two channels.

As mentioned in Section 7.3, the detector used is System C with eight stored vectors (i.e., C8). The delay introduced by the detector is 32 sampling intervals, i.e., l =32. The adaptive filter has 50 taps, i.e., (n+1) = 50, and as such, in all tests with the adaptive filter, the total delay in detection is (n+l) = 81 sampling intervals. The value of g in eqns. 7.2.4 and 7.2.14 is 21 for each of the channels, so that the number of components in the sampled impulse response of each of the channels, and the number of components predicted by the estimator for each of the channels, is, in both cases, 22, i.e., (g+1).

In all tests, the signal-to-noise ratio is taken to be ψ dB, where this is exactly as given in eqns. 4.4.17 and 4.4.18, i.e.,

$$\psi = 10 \log_{10} \left(\frac{E_b}{\frac{1}{2} N_0} \right) \tag{7.6.1}$$

$$= 10 \log_{10} \left(\frac{\overline{s_i^2}}{2\sigma^2} \right) \tag{7.6.2}$$

where E_b, N_0 and σ^2 are as defined in eqns. 4.4.17 and 4.4.18. An average of around 5 x 10^6 data symbols have been involved in the plotting of each curve, giving 95% confidence

limits of less than $\pm$ 0.5 dB.

The notation used in the performance curves is as follows.

PE - perfect estimation assumed.

NF - no filter, i.e., the minimum phase channel is obtained with the aid of the NAG software module, where all the roots that have an absolute value greater than 1.05 are replaced by the complex conjugate of their reciprocals.

MGR - modified gradient estimator.The number that follows this abbreviation, (eg., 30 in MGR30), indicates the signal-to-noise ratio at which the estimator parameters b and θ have been optimized.

Figs. 7.6.1 and 7.6.2 [5] give the performance of the system over Channels 1 and 2, respectively, when perfect estimation is assumed, for the case where the channel is made minimum phase by the previously described ideal method which does not employ the adaptive filter (NF), and for the case where the adaptive filter is employed (WF) to generate the minimum phase response. When the adaptive filter is used with pefect estimation, it operates, at t = iT, on the sampled impulse response of the channel given by Y_{i-n} (see eqn. 7.2.4), rather than on the prediction $Y'_{i-n,i-n-1}$ (eqn. 7.2.14), of Y_{i-n}. Aside of this, the rest of the system operates exactly as described in Section 7.2. It should be noted here that the adaptive filter operates on a different sampled impulse response at every time instant and the tap adjustment is also effected at every time instant.

Since the performance curve without the filter (NF) represents an 'effective upper bound', the performance curve with the filter would give an indication of the degradation caused by the inclusion of the filter. The results here are quite remarkable since it appears that this degradation is quite small, considering the nature and severity of the fading. Indeed, at low signal-to-noise ratios there is absolutely no degradation over either of the channels. This is because, at low signal-to-noise ratios, additive noise is much more of a contributory factor towards the occurrence of errors, than any failure of the adaptive filter to track the roots that have an absolute value greater than 1.05. At high signal-to-noise ratios, the degradation is about 0.6 dB over Channel 1 (Fig. 7.6.1) and 2 dB over Channel 2 (Fig 7.6.2). The degradation is worse in Channel 2 because of the significantly deeper fades exhibited by this channel. At high signal-to-noise ratios, most of the errors occur when the signal is in a deep fade, and this is also when the adaptive filter is most likely to miss some of the roots that have an absolute value greater than 1.05. Thus,

the deeper the fade, the more will be the degradation caused by the inclusion of the adaptive filter. The overall performance of the adaptive filter however, is most encouraging, bearing in mind that Channel 2 exhibits some horrendous fades (Fig. 4.6.2).

When perfect estimation is not assumed, and a modified gradient estimator is employed to give a prediction of the sampled impulse response of the channel as required by the adaptive filter, it was found that the system collapsed, even though the signals were differentially encoded. Thus, it was decided to use an ideal retraining scheme as follows. The data is assumed to occur in blocks of 1000 symbols (2000 bits). After the receipt of every 1000^{th} sample, the prediction vector $Y'_{i-n,i-n-1}$ (eqn. 7.2.14) is made correct by having $Y'_{i-n,i-n-1} = Y_{i-n}$, and the first time derivative of all the components of $Y'_{i-n,i-n-1}$ is set to zero. Thus, at the end of every 1000^{th} sample, the receiver is given knowledge of the channel sampled impulse response, though not of the variation of this sampled impulse response with time. Moreover, all the values in each of the stored vectors in the detector are made correct by having $x_{i-n-h} = s_{i-n-h}$, for $h = 1, 2, ..., l$ (see eqn .7.3.1), and the costs of these stored vectors are reinitialised such that one stored vector has zero cost and the rest have an arbitrarily high value of cost. As such, after every 1000 samples, the detector starts to operate as it did at the beginning of the transmission, with all the values in the stored vectors correct and the costs initialised. This is similar to the situation that would occur in practice, at the end of each training period.

Fig. 7.6.3 shows the performance of the system with the above mentioned retraining, when operating over Channel 1. The estimator used here is the modified gradient estimator whose parameters b and θ have been optimized for a signal-to-noise ratio of 30 dB. It could be seen that there is a degradation, at error rates of 10^{-4}, of some 8 dB relative to the 'upper bound' and 7.5 dB relative to the case where the adaptive filter is used with perfect estimation.

Fig. 7.6.4 shows the performance of the system, again with the above mentioned retraining, but now operating over Channel 2. There are two performance curves with the modified gradient estimator, one where its parameters have been optimized for a signal-to-noise ratio of 30 dB (MGR30), and the other where the parameters have been optimized for a signal-to-noise ratio of 35 dB (MGR35). Although MGR35 is the better system at high signal-to-noise ratios, it is slightly worse than MGR30 at low signal-to-noise ratios, and as such, it is up to the modem designer to choose a signal-to-noise ratio at which the system must work best, and then optimize the parameters of the estimator for that particular signal-to-noise ratio.

In the range of error rates from 10^{-1} to 5×10^{-3}, both MGR30 and MGR35 are, on average, about 9.0 dB worse than the 'upper bound' and about 8.5 dB worse than the case where the adaptive filter is used with perfect estimation. Moreover, there is an irreducible error rate of approximately 1.5×10^{-3} and 7×10^{-4} in MGR30 and MGR35, respectively. As such, the degradation caused by the inclusion of the estimator is worse in Channel 2, than in Channel 1.

These results (especially those in Fig. 7.6.4), on first thought, do not appear to be very encouraging since it was hoped that the estimator would not degrade the performance of the system substantially. However, tests have shown that at error rates of around 10^{-3}, most of the errors occur in just one or two blocks, where these blocks correspond to the times when the signal is in the deepest fades. Moreover, when the noise level is fairly high (say, at error rates of around 5×10^{-2}), once errors start to occur due to the signal encountering a deep fade, they generally continue to occur until the end of the block, which is when retraining is performed once more. This suggests that the performance of the system with the modified gradient estimator is critically dependent on the depth of the fades. As such, since the fading sequences employed are 'worst case' for the given channel parameters, the performance of the system is seen here in a much worse light than the performance that could be expected of the system when operating over a more typical channel.

Furthermore, in the case of the system operating over Channel 1, it can be seen from Fig. 7.6.3, that error-free performance could be achieved, given adequate transmission power. This is a very encouraging result, since, to the authors knowledge, there has been no system proposed which could boast this fact for 4.8 kbit/s HF radio modems which do not use any form of interleaving, ARQ or diversity, and which are designed to handle frequency spreads and time delays similar to those experienced over Channel 1. Moreover, the estimator used here (modified gradient estimator) is basically simple, relative to the, say, Kalman estimator or most derivatives of the Kalman estimator[3]. As such, a more powerful estimator would undoubtedly give a better performance since the degradation has been shown to be mainly due to the inclusion of the estimator. Indeed, tests have shown [6] that an improvement in performance of some 2 dB may be achieved over both Channels 1 and 2, with the use of a slightly more versatile estimator.

The above considerations show the results in Figs. 7.6.3 and 7.6.4 to be encouraging. Indeed, the basic structure of the receiver of the modem, with its particular techniques of detection, estimation and adaptive filtering, suggests a potentially useful design for 4.8 kbit/s HF radio modems, and as such, is worthy of further development.

References

1. **Clark, A.P.,** 'Equalizers for digital modems', Pentech Press, London, 1985.

2. **Clark, A.P., and Hau, S.F.,** 'Adaptive adjustment of receiver for distorted digital signals', IEE Proc. Part F, no.5, pp. 526 - 536, August 1984.

3. **Qureshi, S.U.H.,** 'Adaptive equalization', IEEE Proceedings, vol. 73, no. 9, pp. 1349 - 1387, September 1985.

4. **Eleftherious, E and Falconer, D.D.,** 'Adaptive equalization techniques for HF channels', IEEE Journal on selected areas in Communications, vol. SAC-5, no.2, February 1987.

5. **Jayasinghe S.G., Hariharan, S and Clark, A.P.,** 'An adaptive receiver for a 4.8 kbit/s HF radio modem', IERE Conf. Proc. on Fifth International Conference on Digital Processing of Signals in Communcations, Loughborough, U.K., pp. 239 - 246, September 1988.

6. **Jayasinghe, S.G., Hariharan, S, and Clark, A.P.,** 'Performance of a novel 4.8 kbit/s HF radio modem' submitted to MILCOM '89, Bedford, MA, USA, October 15-18, 1989.

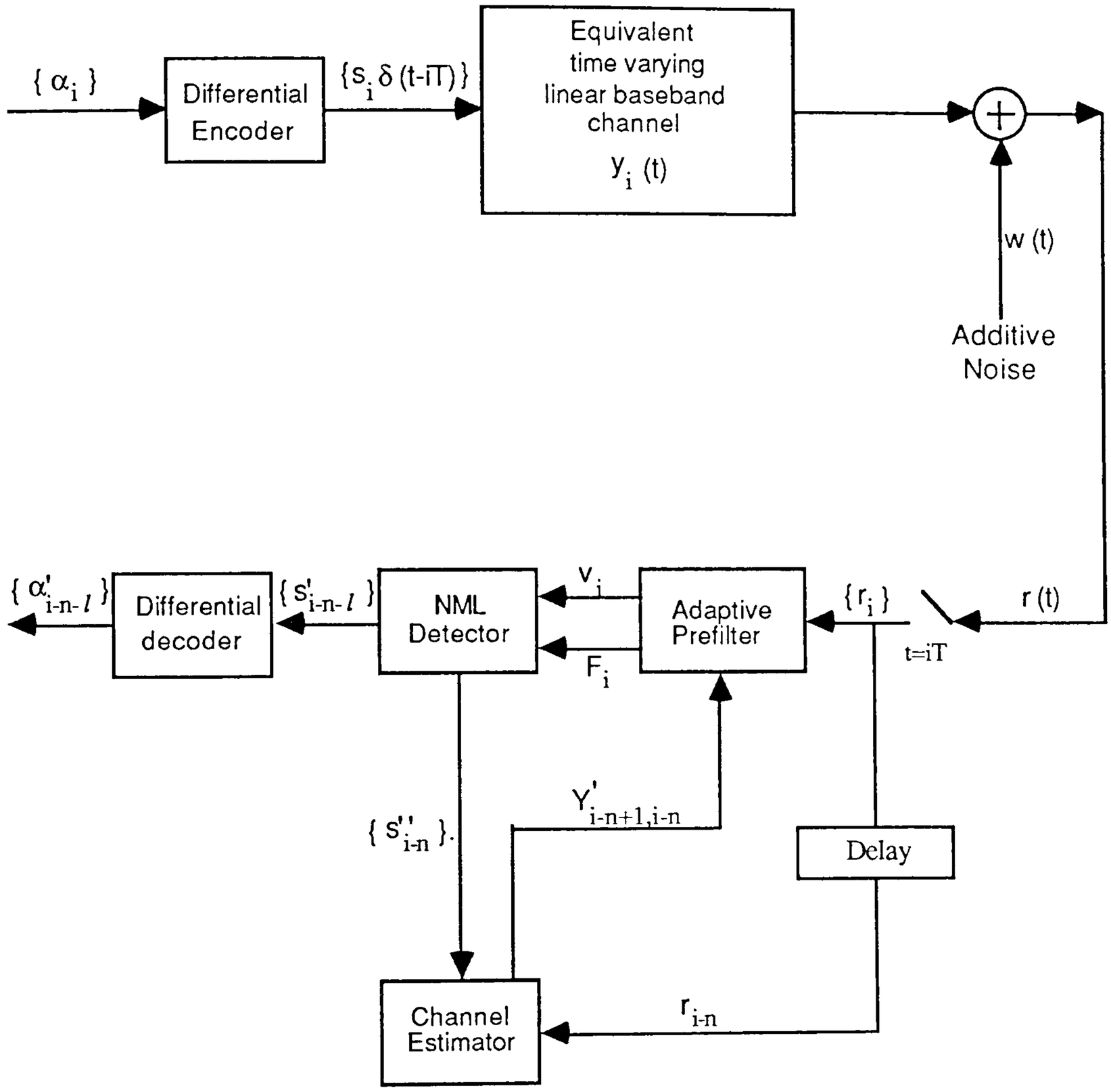


Fig. 7.2.1 Model of data transmission system used in tests in Chapter 7

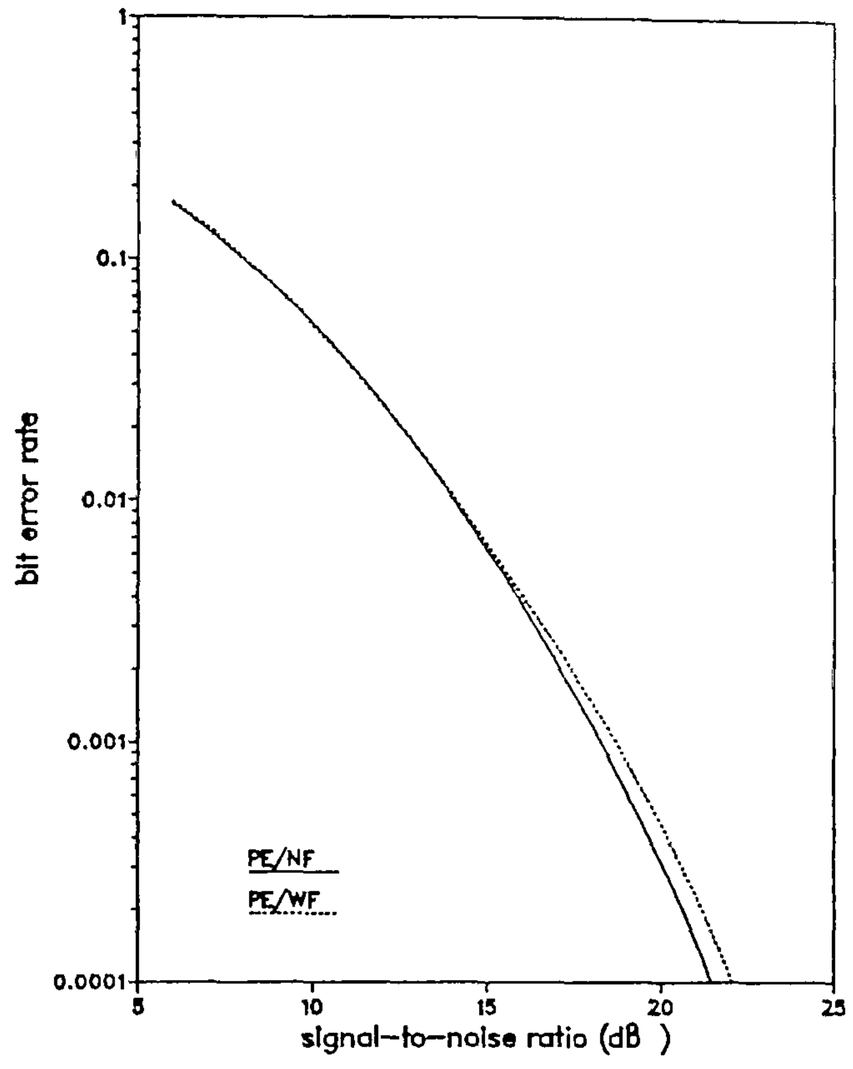


Fig.7.6.1. Performance of adaptive filter over Channel 1 with perfect estimation

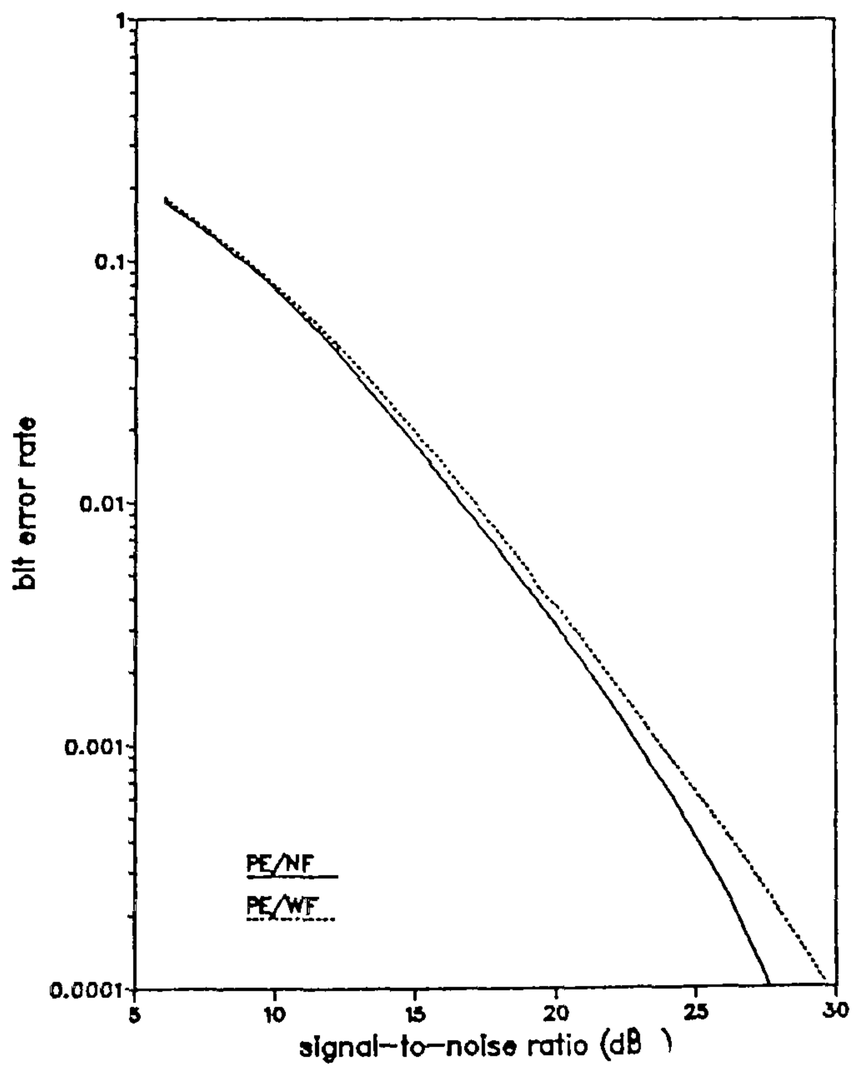


Fig.7.6.2. Performance of adaptive filter over Channel 2 with perfect estimation

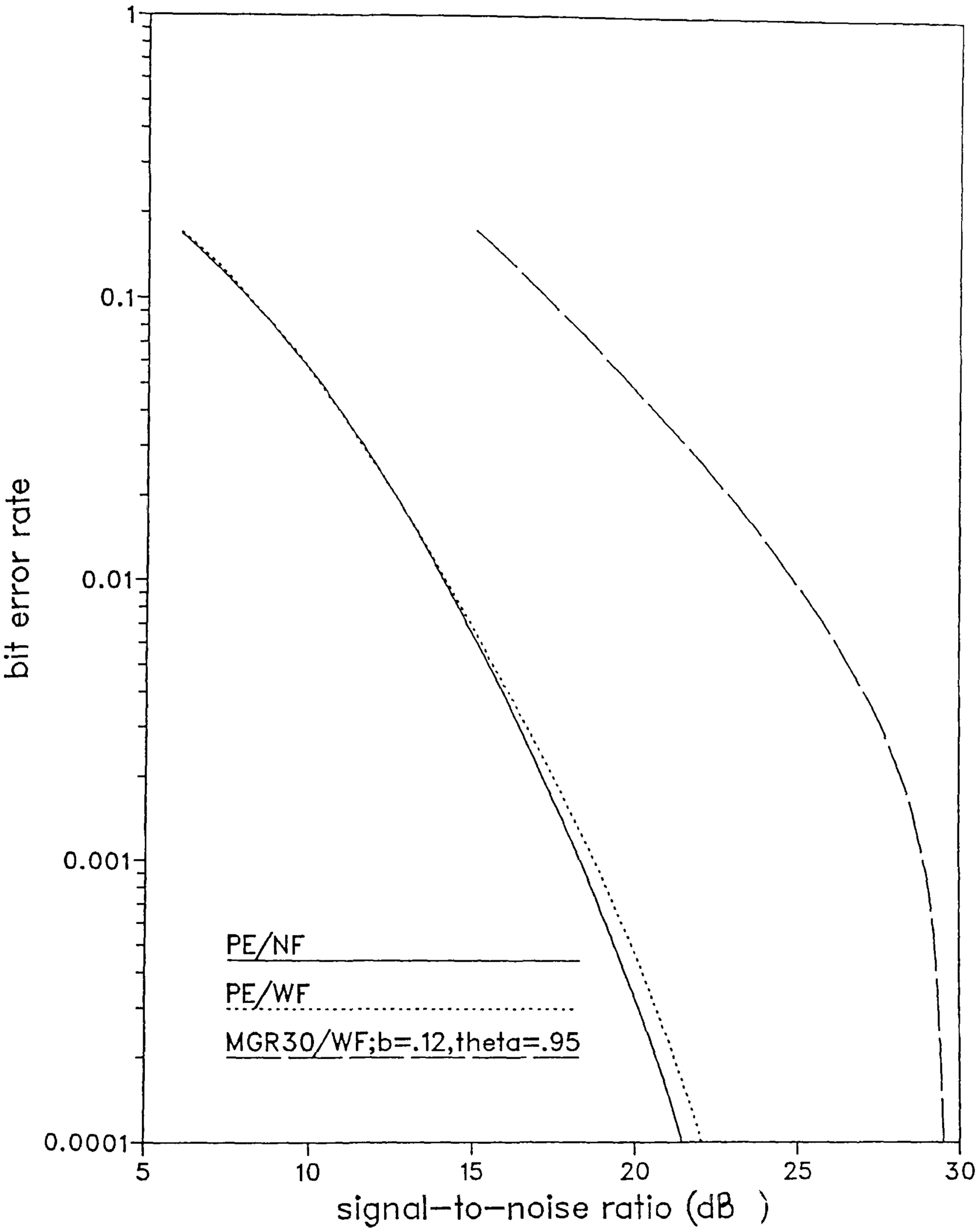


Fig.7.6.3. Performance of system, with retraining over Channel 1

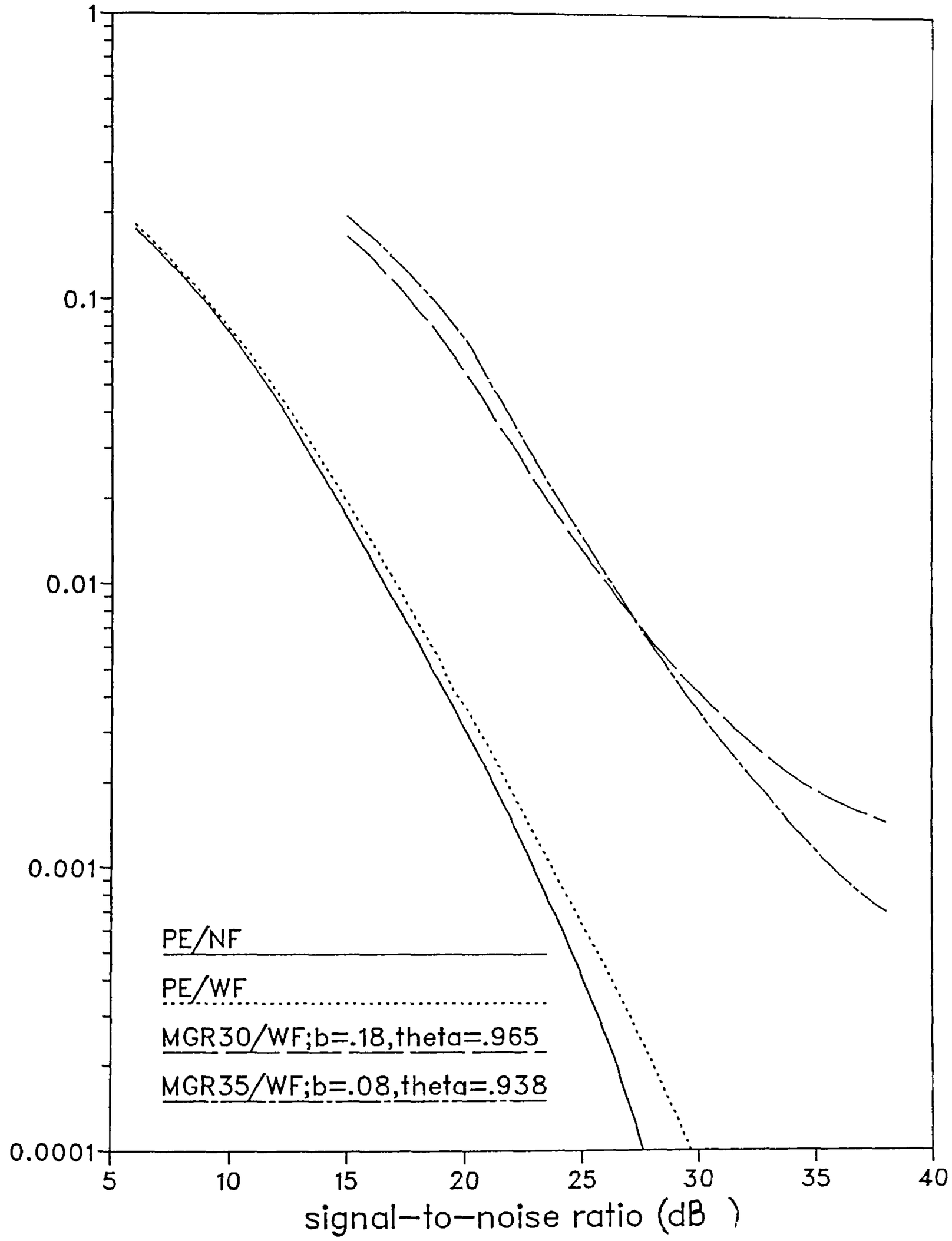


Fig.7.6.4. Performance of system, with retraining over Channel 2

CHAPTER 8

COMMENTS

8.1 SUMMARY OF PROJECT

The research project has been concerned with the development of detection, coding and estimation techniques for modems operating over fading channels. Two types of fading channels have been considered, an HF radio channel (3-30 MHz.) and a fast fading channel with fades of the type that are normally likely to be encountered over cellular land mobile radio (~ 900 MHz.) links.

Various near maximum likelihood detection processes have been tested for operation in a 4.8 kbit/s HF radio modem using 4-level quadrature amplitude modulation. These tests have assumed perfect estimation and the channel is made minimum phase by an ideal method, thus obviating the need for a prefilter to be associated with the detector. The performance of the near maximum likelihood detectors has been compared with that of the corresponding nonlinear equalizer, and the detector that offers the best compromise between performance and complexity is chosen for further tests.

Trellis coded modulation (TCM) has been investigated as a possible forward error correction scheme for HF radio modems. The assumed system is the same 4.8 kbit/s modem mentioned previously, and the tests have been carried out under the same basic assumptions regarding the functions of estimation and prefiltering. 8-PSK coded modulation is tested here, and the code used is a rate-2/3, 8-state, 90° rotationally invariant convolutional code with an asymptotic coding gain of 3.0 dB over the corresponding uncoded modulation scheme, over additive white Gaussian noise channels. The decoders used are two near maximum likelihood decoders that employ soft decisions at the receiver.

Two estimation techniques have been tested for operation in a modem to be used over fast fading channels. Both estimators are variants of the gradient estimator, and are known here as the modified gradient estimator and the unbiased estimator. The system assumed is a 24 kbit/s modem employing 4-level quadrature amplitude modulation. To effect a proper comparison of the estimators, the above tests have been carried out, initially, under the assumption that the transmitted signals are perfectly detected, and the measures of performance used are the mean square error and the mean square normalized error, in the prediction. Another important aim of this investigation has been to study the feasibility of the simultaneous transmission of two bandlimited 4-level QAM signals over two

independently fading channels, where both signals occupy the same frequency band, and no diversity or error correction is used at the receiver to improve the performance. The information rate when two signals are transmitted is, of course, 48 kbit/s. In addition to the estimator performance tests, the effect of errors in detection on the performance of the system is investigated by employing two detectors, a 'simple' detector and a more complicated 'combined' detector. The measure of performance here is the bit error rate (BER) curve. The above tests have been carried out for both the cases of the receipt of one, and two signals over the given frequency band.

Finally, tests have been carried out to determine the performance of the previously mentioned 4.8 kbit/s HF radio modem, when all the functions of detection, estimation and prefiltering are included. The detector used here is the detector chosen from the previous tests, and the estimator employed is the modified gradient estimator. The prefilter is an adaptive linear feedforward transversal filter, whose function is to make the sampled impulse response of the channel and filter, a minimum phase.

Throughout the work carried out in this thesis, all other functions (AGC , symbol timing, etc.) of the modem, except when stated otherwise, have been assumed to be perfect.

8.2 CONCLUSIONS

The results of the performance tests of the detectors have shown that all the near maximum likelihood detectors tested, offer an advantage in terms of their tolerance to noise, over the corresponding nonlinear equalizer. The best of the near maximum likelihood detectors (System 4A16), gains a maximum of about 3 dB over the corresponding nonlinear equalizer. The detector which gives the best compromise between performance and complexity however, is System C with 8 stored vectors, that is, C8.

In the case of trellis coded modulation for the 4.8 kbit/s HF radio modem, the coded systems D and E give a similar performance. However, their overall performance is worse than that of the corresponding uncoded modulation scheme with detector C8. This result may be attributed to the fact that the codes are being tested under the severest of conditions, and the absence of interleaving. It may even be that more powerful codes with larger asymptotic coding gains need to be considered for successful operation over HF radio channels. However, it does appear that the use of TCM may not necessarily provide the expected advantage over uncoded modulation schemes, when operating over channels with deep, frequency selective Rayleigh fading. For the given data transmission system

and channel conditions, uncoded modulation with the detector C8 is the preferred scheme.

The estimator tests of the fast, flat Rayleigh fading channel have shown that the overall performance of the modified gradient estimator is better than that of the unbiased estimator. Moreover, the unbiased estimator is much more complex than the modified gradient estimator, for the case where two signals are received over the given frequency band. Thus, the modified gradient estimator is chosen as the preferred system. When the measure of performance of the estimators is changed from the mean square error in the prediction, to the mean square normalized error in the prediction, the overall conclusions remain unchanged, thereby confirming the validity of the mean square error in the prediction as an appropriate measure of performance.

The performance of the combined detector over the fast fading channel, in the presence of one signal, is within 1.5 dB of the performance of the system with perfect estimation, while the simple detector suffers from an irreducible error rate of about $2x10^{-4}$. Both detectors have been used here with the modified gradient estimator. When two signals are received over the given frequency band, tests have shown that the system collapses completely, unless an appropriate retraining scheme is used. However, even with retraining, both the simple and combined detectors suffer from irreducible error rates in the range $2.5x10^{-2}$ to $7x10^{-3}$. As such, it is concluded that some sort of diversity scheme is necessary for error free operation, in the presence of two received signals, where both signals occupy the same frequency band.

The tests on the 4.8 kbit/s HF radio modem with detector C8 and the prefilter, have shown that the inclusion of the adaptive linear feedforward transversal filter does not degrade the performance of the systems significantly, as long as perfect estimation is assumed. When the receiver has to estimate the channel however, it has been found that the system with the modified gradient estimator requires retraining, in order to prevent a total collapse. When retraining is performed at intervals of 1000 symbols, the degradation caused by the inclusion of the estimator is, on average, 7.4 dB over Channel 1 and 9 dB over Channel 2. Although the system suffers from an irreducible error rate when operating over Channel 2, error free performance can be achieved over Channel 1, given adequate transmission power. Bearing in mind that these results are 'worst case', it is concluded that the proposed system is a potentially useful design for the receiver of a modem transmitting data at 4.8 kbit/s over HF radio channels.

APPENDICES

APPENDIX A

DERIVATION OF 5-POLE BESSEL FILTER

The transfer function, in the s-plane, of a 5-pole Bessel filter could be written as, [1]

$$H(s) = \frac{G_A}{\prod_{i=1}^{5} (s-p_i)} \tag{A.1}$$

where,

G_A = constant dependent on the cutoff frequency
s = Laplacian variable
p_i = s-plane poles

When the filter is normalized to a cutoff (-3dB) frequency of $f_c = 1$Hz., the s-plane poles take on the following values. [1]

$$p_1 = -9.4394 \tag{A.2}$$

$$p_2 , p_3 = -8.6764 \pm j4.5108 \tag{A.3}$$

$$p_4 , p_5 = -6.0173 \pm j9.2434 \tag{A.4}$$

However, the cutoff frequency of interest here, is not 1Hz. but is dependent on the frequency spread that is to be introduced. Thus, the relationship between the cutoff frequency f_c, and the frequency spread f_{sp}, is next obtained.

Consider the process $q_1(t)$. The required power spectrum, as given in eqn. 2.6.1, is,

$$| q_1(f) |^2 = \exp \left(\frac{-f^2}{2f_{rms}^2} \right) \tag{A.5}$$

The cutoff frequency, f_c, is the -3dB (or half power) point and is given by the value of f in eqn. A.5 when $| q_1(f) |^2 \approx 0.5$.

$$0.5 = \exp\left(\frac{-f_c^2}{2f_{rms}^2}\right)$$

$$f_c = 1.1774\, f_{rms} \qquad (A.6)$$

But, from eqn. 2.6.2.,

$$f_{rms} = \frac{f_{sp}}{2} \qquad (A.7)$$

From eqns. A.6 and A.7.,

$$f_c = 0.5887\, f_{sp} \qquad (A.8)$$

Since the two frequency spreads of interest (for Channels 1, 2 and 3) are known, the corresponding cutoff frequencies can be determined. The s-plane poles corresponding to the required cutoff frequencies are next obtained by scaling the values given in eqns. A.2, A.3 and A.4, by the value of f_c. These values are given in Table A.1.

The s-plane poles must next be transformed onto the z-plane in order that the filter may be implemented digitally. Toward this end, the impulse invariance technique [2] is used, where the impulse reponse of the digital filter becomes a sampled version of the impulse response of the analogue filter, and the transfer function becomes,

$$H(z) = \frac{G_D}{\prod_{i=1}^{5} (1 - d_i z^{-1})} \qquad (A.9)$$

where,

G_D = constant

d_i = pole locations in the z-plane

The mapping of the s-plane poles to the z-plane is effected by the transformation,

$$d_i = \exp(p_i T) \qquad (A.10)$$

where T is the sampling interval in seconds. The sequence obtained at the output of this filter is equivalent to the sequence obtained by sampling the waveform at the output of the original filter with a sampling interval of T seconds. The value of T here, was chosen to be 10 ms., for reasons given in Section 2.6.2. The values of the pole locations in the z-plane, for the various frequency spreads, are given in Table A.2, and the value of the constant G_D is adjusted as explained in Section 2.6.2.

The 5-pole Bessel filter with z-plane poles d_i, (i = 1 to 5), is realized as shown in Fig. A.1, where the tap gain co-efficients are given by c_i, (i = 1 to 5). [3] This is a cascade of two 2-pole sections and a single 1-pole section. The poles in each 2-pole section are complex conjugates and are, d_2 and d_3, and, d_4 and d_5. The single 1-pole section has a real pole d_1. From Fig. A.1, it could be seen that the transfer function of the first 2-pole section is given by,

$$T_T(f) = \frac{1}{1 + c_1 z^{-1} + c_2 z^{-2}} \qquad (A.11)$$

If this section is allowed the complex conjugate pole pair d_2 and d_3, an equivalent representation of this transfer function would be,

$$T_T(f) = \frac{1}{(1 - d_2 z^{-1})(1 - d_3 z^{-1})} \qquad (A.12)$$

$$T_T(f) = \frac{1}{1 - (d_2 + d_3) z^{-1} + (d_2 d_3) z^{-2}} \qquad (A.13)$$

Fron eqns. A.11 and A.13, the relationship between the tap gain co-efficients and the z-plane poles could be obtained, i.e.,

$$c_1 = -(d_2 + d_3) \qquad (A.14)$$

$$c_2 = d_2 d_3 \qquad (A.15)$$

Similarly, if the complex conjugate pole pair d_4 and d_5 were assigned to the second 2-pole section,

$$c_3 = -(d_4 + d_5) \qquad (A.16)$$

$$c_4 = d_4 d_5 \tag{A.17}$$

Finally, the transfer function of the single 1-pole section is given by,

$$T_s(f) = \frac{1}{1 + c_5 z^{-1}} \tag{A.18}$$

This section is allowed the real pole d_1, and an equivalent representation of this transfer function is obtained as,

$$T_s(f) = \frac{1}{1 - d_1 z^{-1}} \tag{A.19}$$

From eqns. A.18 and A.19,

$$c_5 = - d_1 \tag{A.20}$$

Thus, the tap gain co-efficients of the filter are derived (eqns. A.14 - A.17 and eqn. A.20), and their numerical values are given in Table A.3.

References - Appendix A

1. **Kuo, F.F.**, 'Network analysis and synthesis', John Wiley & Sons Inc., 1966.

2. **Oppenheim, A.V. and Schafer, R.W.**, 'Digital signal processing',1975.

3. **McVerry, F.**, 'High speed data transmission over HF radio links', Ph.D. thesis, Loughborough University of Technology, 1982.

f_{sp} Hz.	f_c Hz.	s - plane poles		
		p_1	p_2 and p_3	p_4 and p_5
1	0.5887	-5.56	-5.11 ± j2.67	-3.54 ± j5.44
2	1.1774	-11.11	-10.22 ± j5.31	-7.09 ± j10.88
125	73.5875	–694.63	-638.48 ± j331.94	-442.80 ± j680.25

Table A.1 s-plane poles of Bessel filter

f_{sp} Hz.	z - plane poles		
	d_1	d_2 and d_3	d_4 and d_5
1	0.9459	0.9498 ± j0.0252	0.9638 ± j0.0525
2	0.8948	0.9016 ± j0.0479	0.9261 ± j0.1019
125	0.5605	0.5651 ± j0.1604	0.5832 ± j0.3713

Table A.2 z-plane poles of Bessel filter

f_{sp} Hz.	tap gain co-efficients				
	c_1	c_2	c_3	c_4	c_5
1	-1.8996	0.9028	-1.9270	0.9316	-0.9459
2	-1.8032	0.8152	-1.8520	0.8678	-0.8948
125	-1.1300	0.3450	-1.1670	0.4780	-0.5610

Table A.3 Coefficients of Bessel filter

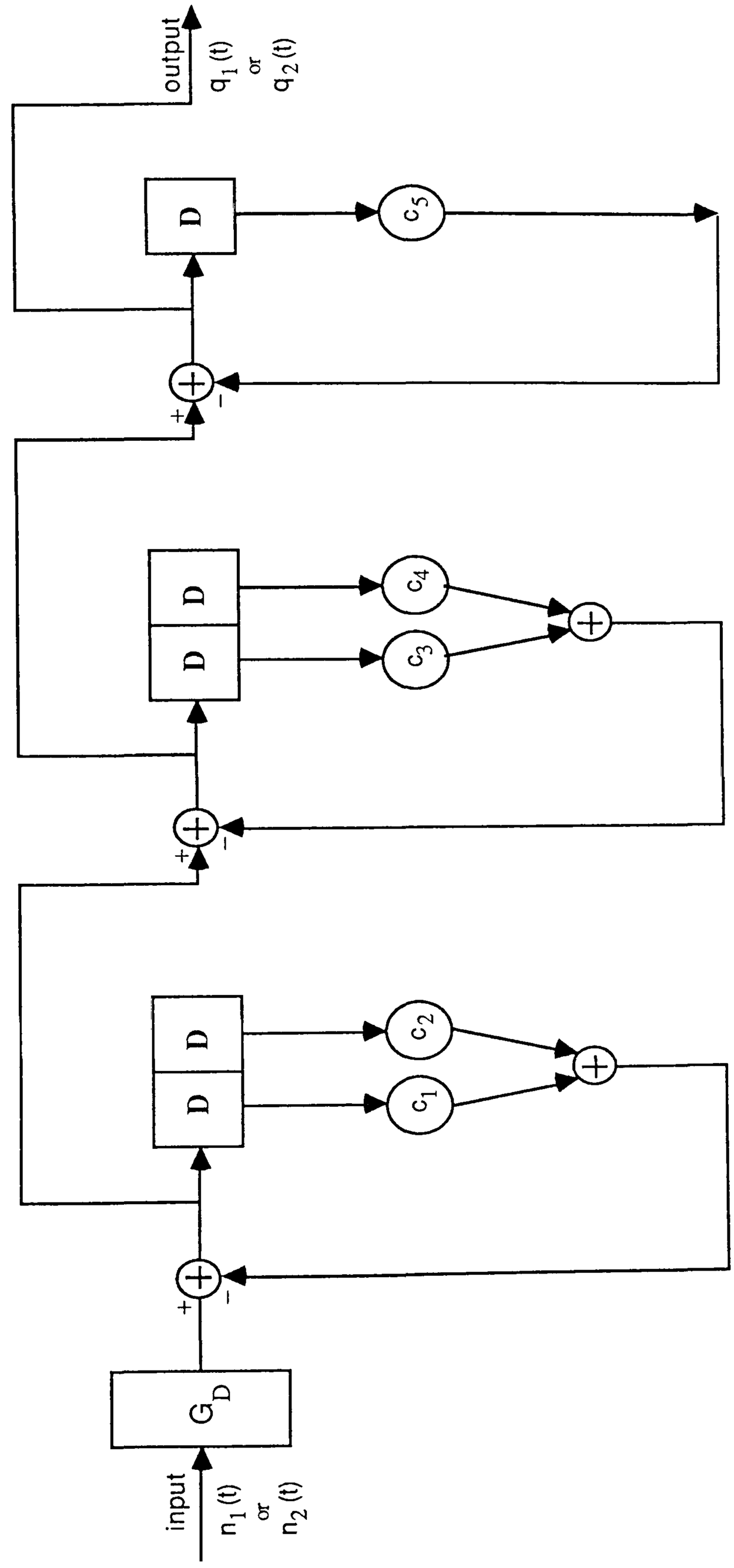


Fig. A.1 Block diagram of 5-pole Bessel filter

APPENDIX B

DIFFERENTIAL ENCODING AND DECODING

Differential encoding of the transmitted message is carried out as follows. The transmitted stream of binary digits $\{\alpha_i\}$ (Fig. 4.4.1), are first divided, consecutively, into groups of two. To simplify the terminology, let the two binary digits in the i^{th} group be represented by $\alpha_{1,i}$ and $\alpha_{2,i}$, as shown in Fig. B.1. These two binary digits are next encoded into the two binary digits $\beta_{1,i}$ and $\beta_{2,i}$, (with the aid of $\beta_{1,i-1}$ and $\beta_{2,i-1}$) according to Table B.1. Finally, $\beta_{1,i}$ and $\beta_{2,i}$ are recoded into the complex valued signal s_i, according to Table B.3 and as illustrated in Fig. B.3, where s_i is depicted as $s_{1,i} + js_{2,i}$.

In order to obtain the correct detected values of the $\{\alpha_i\}$, the encoding process must be reversed, (or, decoding should be carried out), at the output of the detector in Fig. 4.4.1. This decoding process is illustrated in Fig. B.2. The detected binary digits $\beta'_{1,i}$ and $\beta'_{2,i}$, are first determined from the detected data symbol values $\{s_i'\}$, again according to the relationships given in Table B.3 and Fig. B.3, the difference being that the values used now are the corrseponding detected values. $\beta'_{1,i}$ and $\beta'_{2,i}$ are used in the decoder, along with $\beta'_{1,i-1}$ and $\beta'_{2,i-1}$, to give the detected values of $\alpha_{1,i}$ and $\alpha_{2,i}$, namely, $\alpha'_{1,i}$ and $\alpha'_{2,i}$. This is done according to Table B.2.

It can be seen, from Table B.1 and Fig. B.3, that the signal corresponding to the group of digits $\alpha_{1,i}\,\alpha_{2,i}$, represents the difference in phase between the two signals corresponding to the two groups of digits $\beta_{1,i}\,\beta_{2,i}$ and $\beta_{1,i-1}\,\beta_{2,i-1}$. As such, the phase of the signal corresponding to $\beta_{1,i}\,\beta_{2,i}$, is equal to the sum of the two phases of the signals corresponding to $\alpha_{1,i}\,\alpha_{2,i}$ and $\beta_{1,i-1}\,\beta_{2,i-1}$. Thus, given the two groups of digits $\alpha_{1,i}\,\alpha_{2,i}$ and $\beta_{1,i-1}\,\beta_{2,i-1}$, the group of digits $\beta_{1,i}\,\beta_{2,i}$ could readily be obtained by referring to Fig. B.3. Moreover, phase shifts of multiples of 90° in the received signal would no longer lead to prolonged error bursts in the detected binary digits $\{\alpha'_i\}$, since the transmitted binary digits now represent 'phase differences'. It should be noted here, that in addition to differential coding, the signal is Gray coded (Fig. B.3), such that adjacent values of s_i

differ only in one digit. This, of course, reduces further, the probability of error in the detected data signals.

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

Table B.1 **Differential encoding of binary digits for 4-QAM system**

β'_{i-1}		β'_i		α'_i	
$\beta'_{1,i-1}$	$\beta'_{2,i-1}$	$\beta'_{1,i}$	$\beta'_{2,i}$	$\alpha'_{1,i}$	$\alpha'_{2,i}$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

Table B.2 **Differential decoding of binary digits for 4-QAM system**

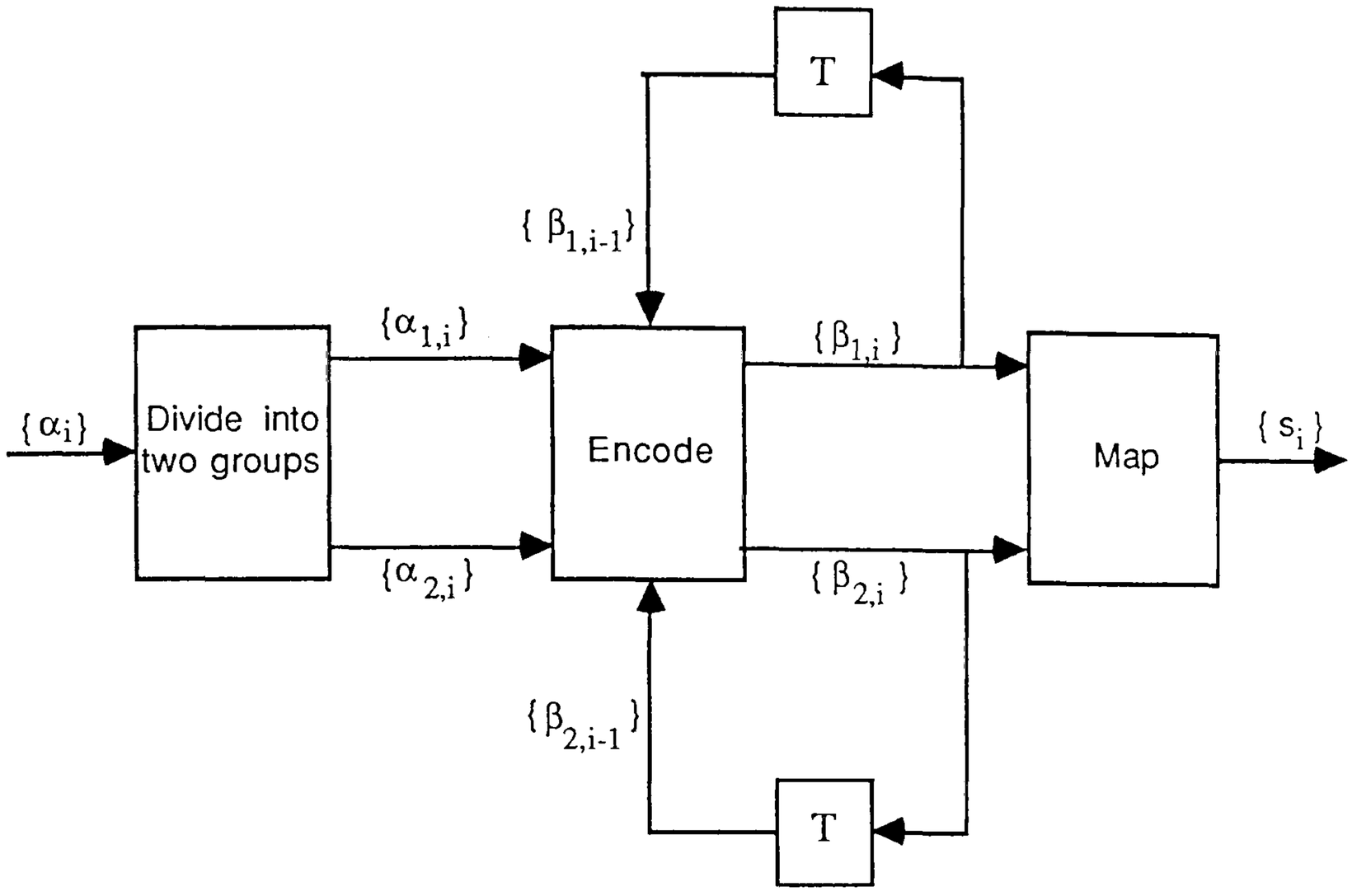


Fig. B.1 The differential encoding process

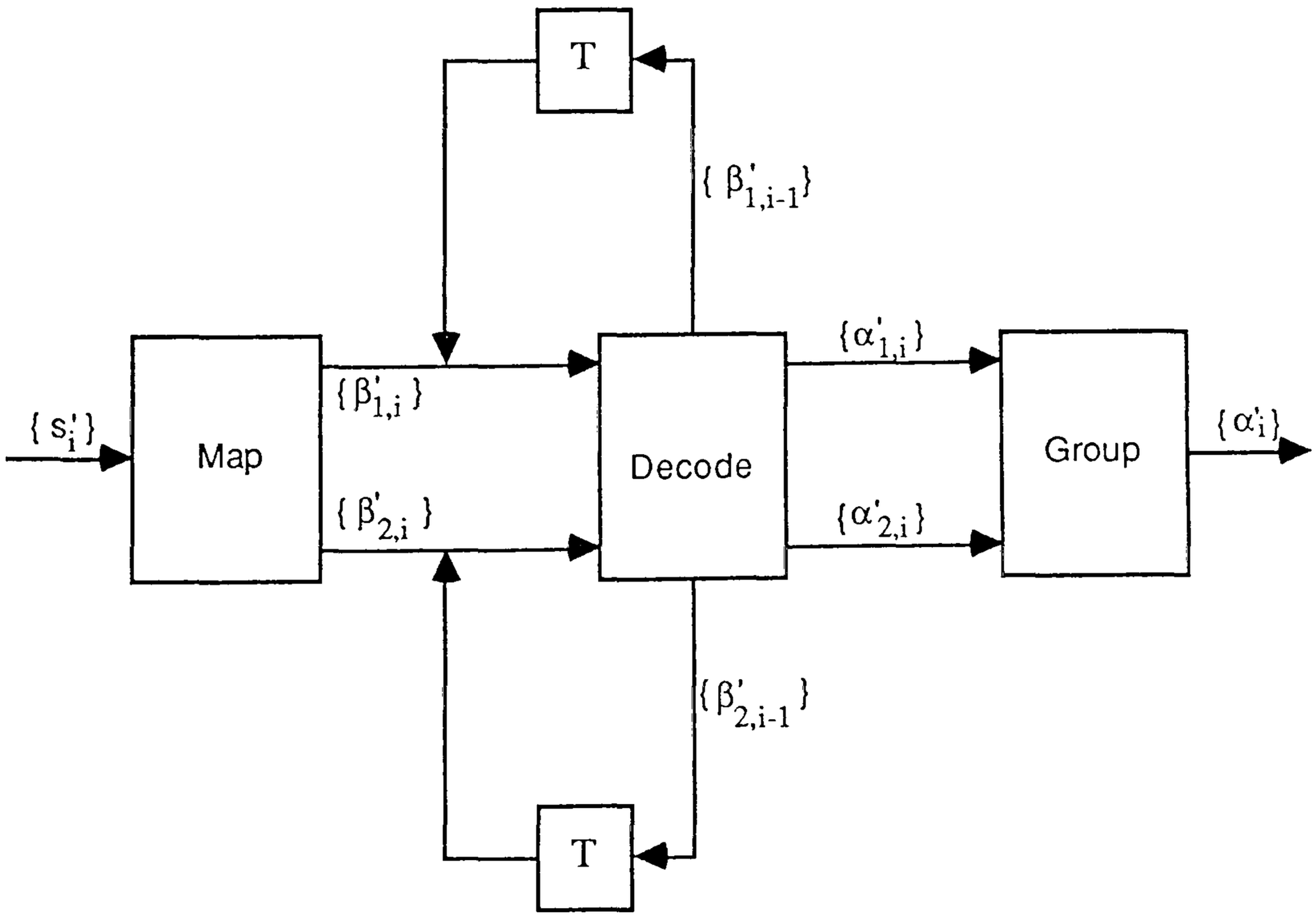


Fig. B.2 The differential decoding process

$\beta_{1,i}$	$\beta_{2,i}$	s_i
0	0	1 + j1
0	1	-1 + j1
1	0	1 - j1
1	1	-1 - j1

Table B.3 Encoding relationship for $\{s_i\}$

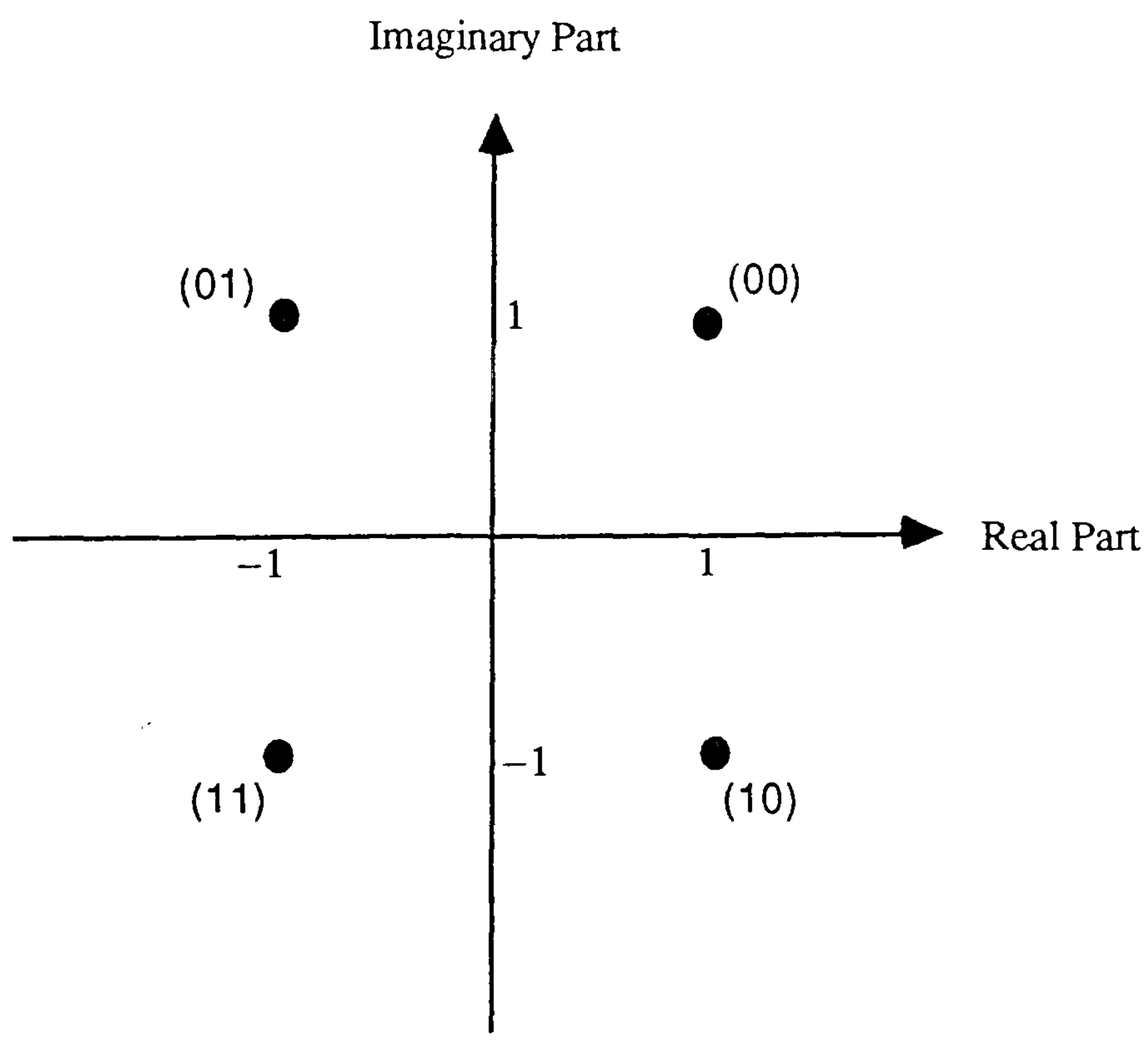


Fig B.3 The relationship in Table B.3 depicted in the complex number plane

APPENDIX C

SIGNAL-TO-NOISE RATIO CALCULATIONS

In the simulated data transmission system model given in Figs. 4.4.1 and 4.4.2, the signal-to-noise ratio is defined as

$$\psi = 10 \log_{10} \left(\frac{E_b}{\frac{1}{2} N_0} \right) \qquad (C.1)$$

where E_b is the average transmitted energy per bit at the output of the transmitter filter A in Fig. 4.4.2, and, $N_o/2$ is the two-sided power spectral density of the real valued noise function n(t) in Fig. 3.3.1. The complex valued noise function w(t), in Fig. 4.4.1 is, of course, derived from the real valued noise function n(t) as explained in Section 3.3. It is now necessary to express the signal-to-noise ratio in terms of quantities that are readily obtainable from the simulation process.

From eqn. 3.3.40, E_b could be expressed as

$$E_b = \frac{\overline{s_i^2}}{\log_2 m} \int_{-\infty}^{\infty} |A(f)|^2 \, df \qquad (C.2)$$

where s_i^2 is the expected value of $|s_i|^2$, m is the number of levels of the transmitted signal, and |A(f)| is the absolute value of the transfer function of the transmitter filter A. Noting that, for a 4-level QAM signal, m=4, and using Parseval's theorem [1],

$$E_b = \frac{\overline{s_i^2}}{2} \int_{-\infty}^{\infty} |a(t)|^2 \, dt \qquad (C.3)$$

It is well known [2,3] that if X(f) is a bandlimited spectrum whose inverse Fourier transform is the continous waveform x(t), and if x(t) is sampled at a rate greater than the Nyquist rate, to give the samples $\{x_h\}$, then,

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = T_1 \sum_{h=-\infty}^{\infty} |x_h|^2 \qquad (C.4)$$

where T_1 is the sampling period. This result is intuitively rather obvious as well, since the integral on the left hand side of eqn. C.4 represents an area, and for the right hand side to equal this, each component of the summation must be multiplied by the time period seperating the samples. Of course, the lesser the time period, the more accurate the result, with exact correspondence between these continous and discrete time domain functions being obtained when the sampling rate, i.e. 1/T, is at or greater than the Nyquist rate. Using the result in eqn. C.4, E_b can be expressed, from eqn. C.3, as

$$E_b = \frac{\overline{s_i^2}}{2} \frac{T}{2} \sum_{k=-\infty}^{\infty} |a_{1,k}|^2 \qquad (C.5)$$

where the $\{a_{1,k}\}$ are the sample values of the filter A, obtained at 4800 samples/s (Table 4.4.1), and 1/T is the baud rate of the transmission system, i.e. T=1/2400.

It is next attempted to express the power spectral density ($N_o/2$) of the noise waveform n(t), in terms of the variance, $(\sigma_w)^2$, of the complex valued noise waveform w(t). Toward this end consider eqn. 3.3.37, that is,

$$\sigma_w^2 = N_0 \int_{-f_c}^{+f_c} |C(f+f_c)|^2 \, |B'(f)|^2 \, df \qquad (C.6)$$

$(\sigma_w)^2$, here, is the variance of the continous time noise waveform w(t), that is shown in Figs. 3.3.5 and 4.4.1, and C(f) and B'(f) are the transfer functions of the filters C and B', that are shown in Fig. 3.3.1. However, from eqn. 3.3.25,

$$b(t) = [\, d(t) * c(t) \,] \, e^{-j2\pi f_c t} * b'(t) \qquad (C.7)$$

giving rise to,

$$|B(f)|^2 = |D(f+f_c)|^2 \, |C(f+f_c)|^2 \, |B'(f)|^2 \qquad (C.8)$$

where D(f) is the transfer function of the radio receiver filter D in Fig. 3.3.1, and B(f) is the transfer function of the filter B in Figs. 3.3.5 and 4.4.2. But, from the characteristics of the filters (Figs. 4.4.3 - 4.4.5), it can been seen that the radio filters have a much wider bandwidth, compared to the modem filters, and as such, D(f) could be assumed to represent an ideal voice channel over the passband of the modem filters C and B'. Therefore, eqn. C.8 can be approximated by,

$$|B(f)|^2 = |C(f+f_c)|^2 \, |B'(f)|^2 \qquad (C.9)$$

Now, from eqns. C.6 and C.9,

$$\sigma_w^2 = N_0 \int_{-f_c}^{f_c} |B(f)|^2 \, df \qquad (C.10)$$

and using Parseval's theorem, eqn. C.10 could be rewritten as

$$\sigma_w^2 = N_0 \int_{-\infty}^{\infty} |b(t)|^2 \, dt \qquad (C.11)$$

Using the result in eqn. C.4, $(\sigma_w)^2$ can be expressed, from eqn. C.11, as

$$\sigma_w^2 = N_0 \frac{T}{2} \sum_{k=-\infty}^{\infty} |b_k|^2 \qquad (C.12)$$

where the $\{b_k\}$ are the sample values of the filter B in Fig. 4.4.2, obtained at 4800 samples/s (Table 4.4.1), and T/2 is, of course, the sampling period of 1/4800 secs. From eqn. C.12,

$$\frac{N_0}{2} = \frac{\sigma_w^2}{T \sum_{k=-\infty}^{\infty} |b_k|^2} \qquad (C.13)$$

From eqns. C.5 and C.13,

$$\frac{E_b}{\frac{1}{2}N_0} = \frac{\overline{s_i^2}\,T}{4}\sum_{k=-\infty}^{\infty}|a_{1,k}|^2\,\frac{T}{\sigma_w^2}\sum_{k=-\infty}^{\infty}|b_k|^2$$

$$= \frac{\overline{s_i^2}}{4\sigma_w^2}\,T^2\sum_{k=-\infty}^{\infty}|a_{1,k}|^2\sum_{k=-\infty}^{\infty}|b_k|^2 \qquad \text{(C.14)}$$

Now, in order for the signal-to-noise ratio calculation to be made easier, it is necessary to eliminate the factor T^2 in eqn. C.14. This is easiest done by normalizing the filters A and B such that the sum of the squares of their sample values become equal, numerically, to their sampling rates, that is,

$$\sum_{k=-\infty}^{\infty}|a_{1,k}|^2 = \sum_{k=-\infty}^{\infty}|b_k|^2 = \frac{2}{T} \qquad \text{(C.15)}$$

Substituting the values from eqn. C.15, in eqn. C.14, gives

$$\frac{E_b}{\frac{1}{2}N_0} = \frac{\overline{s_i^2}}{\sigma_w^2} \qquad \text{(C.16)}$$

Finally, it is necessary to establish a relationship between $(\sigma_w)^2$ and $2\sigma^2$, where the former is the variance of the continuous time noise wavefrom w(t), and the latter is the total variance of the real and imaginary parts of the discrete noise samples generated from the software module G05DDF, obtained from the Numerical Algorithm Group (NAG) library. For the particular discrete time noise model assumed, it can be shown [4] that these two variances are related by the equation,

$$\sigma_w^2 = 2\sigma^2\sum_{k=-\infty}^{\infty}|b_k|^2 \qquad \text{(C.17)}$$

If now, $\sum|b_k|^2 = 1$, the two variances would be numerically equal. However, since this is not the case, the discrete time noise samples obtained via the software module G05DDF, are passed through a scaled version of the receiver filter B, whose samples are given by $\{b_k''\}$, and where the scaling is carried out such that $\sum|b_k''|^2 = 1$. This would introduce

the correct correlation to the noise samples, while maintaining the relationship,

$$\sigma_w^2 = 2\sigma^2 \tag{C.18}$$

Then, from eqns. C.16 and C.18,

$$\frac{E_b}{\frac{1}{2}N_0} = \frac{\overline{s_i^2}}{2\sigma^2} \tag{C.19}$$

and from eqns. C.1 and C.19, the signal-to-noise ratio could now be represented as

$$\psi = 10 \log_{10} \left(\frac{\overline{s_i^2}}{2\sigma^2} \right) \tag{C.20}$$

For the case of the more idealistic model assumed in Chapter 6, the result is the same. Consider Fig. 6.3.1 and Fig. 3.2.7. The noise function w(t) in Fig. 6.3.1 is that in the equivalent baseband model and, as such, is similar to w(t) in Fig. 3.2.7. From eqn. 3.2.23 it can be seen that the autocorrelation function of each of the real or imaginary part of w(t) is given by,

$$R_w'(\tau) = \frac{N_0}{2} \int_{-\infty}^{\infty} |Z_2(f)|^2 \exp(j2\pi f\tau)\, df \tag{C.21}$$

where $Z_2(f)$ is the overall filtering that is carried out at the receiver (Fig. 6.3.1), which includes the bandpass and lowpass filtering in the corresponding bandpass system model (Fig. 3.2.1). Thus, the variance of the real or imaginary part of w(t) is

$$(\sigma_w')^2 = \frac{N_0}{2} \int_{-\infty}^{\infty} |Z_2(f)|^2\, df \tag{C.22}$$

Since the filtering in the model developed in Chapter 6 is assumed to be equally divided at the transmitter and receiver, considering the first transmission path, TX_0, in Fig. 6.3.1.,

$$|Z_0(f)| = |Z_2(f)| \tag{C.23}$$

A similar result is obtained when the second transmission path, TX_1, is considered. Thus, if the transfer function of the transmitter and receiver filters in cascade is H(f), then,

$$|H(f)| = |Z_0(f)|\,|Z_2(f)| \tag{C.24}$$

Substituting the value of eqn. C.23 in eqn. C.24,

$$|H(f)| = |Z_2(f)|^2 \tag{C.25}$$

However, for the optimum design of the transmitter and receiver filters (as in the case assumed in the tests in Chapter 6), another convenient proviso is that

$$\int_{-\infty}^{\infty} |H(f)|\,df = 1 \tag{C.26}$$

Thus, from eqns. C.22, C.25 and C.26,

$$(\sigma_w')^2 = \frac{N_0}{2} \tag{C.27}$$

Consequently, the variance of the real or imaginary part of the noise sample w_i (eqn. 6.3.1) is

$$\sigma^2 = (\sigma_w')^2 = \frac{N_0}{2} \tag{C.28}$$

Thus from eqns. C.1 and C.28, the signal-to-noise ratio of the system could alternatively be expressed as

$$\psi = 10 \log_{10} \left(\frac{E_b}{\sigma^2}\right) \tag{C.29}$$

It is also appropriate here, to discuss the statistical independence of the

Gaussian random variables w_i (eqn. 6.3.1). From eqns. C.21 and C.25,

$$R_w'(\tau) = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)| \exp(j2\pi f\tau)\, df \tag{C.30}$$

$$= \frac{N_0}{2} h(\tau) \tag{C.31}$$

where $h(\tau)$ is the corresponding time domain response. For zero intersymbol interference filtering (i.e., ideal Nyquist, raised cosine, etc),

$$h(iT) = 0 \tag{C.32}$$

for any nonzero integer i, where T here is the symbol period. From eqns. C.31 and C.32 it could be seen that,

$$R_w'(iT) = \frac{N_0}{2} h(iT) = 0 \tag{C.33}$$

for any nonzero integer i. Now, the sampling instants for any two noise samples, w_k and w_l, are separated by a multiple of T secs., and the noise samples have zero mean. Thus, the real or imaginary part of any two noise samples w_k and w_l (and hence, the two noise samples themselves), are uncorrelated, and therefore, are also statistically independent Gaussian random variables.

References - Appendix C

1. **Clark, A.P.,** 'Principles of digital data transmission', Pentech Press, 1983.

2. **Oppenheim, A.V., and Shafer, R.W.,** 'Digital signal processing', Prentice Hall, 1975.

3. **Schwartz, M.,** 'Information transmission, modulation, and noise', McGraw Hill, 1981.

4. **Najdi, H.,** 'Digital data transmission over voice channels', Ph.D. thesis, Loughborough University of Technology, 1982.

APPENDIX D

UNBIASED ESTIMATES

D.1 Unbiased estimate of a fast fading channel

The estimator here assumes that [1],

$$y_{0,i+1} - y_{0,i} = y_{0,i} - y_{0,i-1} \approx y''_{0,i,i-1} \tag{D.1}$$

so that

$$2y_{0,i} = y_{0,i+1} + y_{0,i-1} \tag{D.2}$$

and similarly for $y_{1,i}$. To estimate $y_{0,i}$, it is necessary to remove $y_{1,i}$ from the received samples. This can be achieved by operating on the three received sample r_{i+1}, r_i and r_{i-1}. Before starting the estimation process, it is necessary for the estimator to know all the data symbols involved in the three samples. The data symbols $s_{0,i+1}$ and $s_{1,i+1}$ are detected after the receipt of r_{i+1}, using the two-step predictions $y'_{0,i+1,i-1}$ and $y'_{1,i+1,i-1}$ of $y_{0,i+1}$ and $y_{1,i+1}$, respectively. Correct detection of all data symbols is assumed, as before.

The three samples r_{i+1}, r_i and r_{i-1}, are given by

$$r_{i+1} = s_{0,i+1}\, y_{0,i+1} + s_{1,i+1}\, y_{1,i+1} + w_{i+1} \tag{D.3}$$

$$r_i = s_{0,i}\, y_{0,i} + s_{1,i}\, y_{1,i} + w_i \tag{D.4}$$

$$r_{i-1} = s_{0,i-1}\, y_{0,i-1} + s_{1,i-1}\, y_{1,i-1} + w_{i-1} \tag{D.5}$$

From eqns. D.3, D.4 and D.5.,

$$s^{-1}_{1,i+1}\, r_{i+1} = s^{-1}_{1,i+1}\, s_{0,i+1}\, y_{0,i+1} + y_{1,i+1} + s^{-1}_{1,i+1}\, w_{i+1} \tag{D.6}$$

$$2\, s_{1,i}^{-1}\, r_i = 2\, s_{1,i}^{-1}\, s_{0,i}\, y_{0,i} + 2\, y_{1,i} + 2\, s_{1,i}^{-1}\, w_i \qquad \text{(D.7)}$$

$$s_{1,i-1}^{-1}\, r_{i-1} = s_{1,i-1}^{-1}\, s_{0,i-1}\, y_{0,i-1} + y_{1,i-1} + s_{1,i-1}^{-1}\, w_{i-1} \qquad \text{(D.8)}$$

From eqns. D.6, D.7 and D.8,

$$\begin{aligned} 2\, s_{1,i}^{-1}\, r_i - s_{1,i+1}^{-1}\, r_{i+1} - s_{1,i-1}^{-1}\, r_{i-1} \\ = 2\, s_{1,i}^{-1}\, s_{0,i}\, y_{0,i} - s_{1,i+1}^{-1}\, s_{0,i+1}\, y_{0,i+1} - s_{1,i-1}^{-1}\, s_{0,i-1}\, y_{0,i-1} \\ + 2\, y_{1,i} - y_{1,i+1} - y_{1,i-1} \\ + 2\, s_{1,i}^{-1}\, w_i - s_{1,i+1}^{-1}\, w_{i+1} - s_{1,i-1}^{-1}\, w_{i-1} \end{aligned} \qquad \text{(D.9)}$$

But, for degree-1 prediction [2] ,

$$y_{0,i+1} \approx y_{0,i} + y''_{0,i,i-1} \qquad \text{(D.10)}$$

and

$$y_{0,i-1} \approx y_{0,i} - y''_{0,i,i-1} \qquad \text{(D.11)}$$

Also, from eqn. D.2,

$$2\, y_{1,i} = y_{1,i+1} + y_{1,i-1} \qquad \text{(D.12)}$$

Substituting the values of eqns. D.10, D.11 and D.12, in eqn. D.9,

$$\begin{aligned} 2\, s_{1,i}^{-1}\, r_i - s_{1,i+1}^{-1}\, r_{i+1} - s_{1,i-1}^{-1}\, r_{i-1} \\ \approx \left(2\, s_{1,i}^{-1}\, s_{0,i} - s_{1,i+1}^{-1}\, s_{0,i+1} - s_{1,i-1}^{-1}\, s_{0,i-1} \right) y_{0,i} \end{aligned}$$

$$- \left(s_{1,i+1}^{-1} s_{0,i+1} - s_{1,i-1}^{-1} s_{0,i-1} \right) y''_{0,i,i-1}$$

$$+ \left(2 s_{1,i}^{-1} w_i - s_{1,i+1}^{-1} w_{i+1} - s_{1,i-1}^{-1} w_{i-1} \right) \qquad \text{(D.13)}$$

Eqn. D.13 can be expressed as,

$$r_{0,i} \approx a_{0,i} y_{0,i} + u_{0,i} \qquad \text{(D.14)}$$

Eqn. D.14 is the same as eqn. 6.4.20, albeit with its terms redefined as follows.

$$r_{0,i} = 2 s_{1,i}^{-1} r_i - s_{1,i+1}^{-1} r_{i+1} - s_{1,i-1}^{-1} r_{i-1}$$

$$+ \left(s_{1,i+1}^{-1} s_{0,i+1} - s_{1,i-1}^{-1} s_{0,i-1} \right) y''_{o,i,i-1} \qquad \text{(D.15)}$$

$$a_{0,i} = 2 s_{1,i}^{-1} s_{0,i} - s_{1,i+1}^{-1} s_{0,i+1} - s_{1,i-1}^{-1} s_{0,i-1} \qquad \text{(D.16)}$$

$$u_{0,i} = 2 s_{1,i}^{-1} w_i - s_{1,i+1}^{-1} w_{i+1} - s_{1,i-1}^{-1} w_{i-1} \qquad \text{(D.17)}$$

An unbiased estimate of $y_{0,i}$ is now given by $x_{0,i}$, where,

$$x_{0,i} \approx a_{0,i}^{-1} r_{0,i} \approx y_{0,i} + a_{0,i}^{-1} u_{0,i} \qquad \text{(D.18)}$$

with $r_{0,i}$, $a_{0,i}$ and $u_{0,i}$ as given in eqns. (D.15-D.17). The error in the one-step prediction $y'_{0,i,i-1}$, is then taken as (see eqn. 6.4.25),

$$\varepsilon_{0,i} = b \, |a_{0,i}| \, (x_{0,i} - y'_{0,i,i-1}) \qquad \text{(D.19)}$$

where b is, as before, a small positive constant. Finally, the degree-1 least squares fading memory prediction algorithm (degree-1 polynomial filter in Table 6.4.1) is used to obtain the one-step prediction, $y'_{0,i+1,i}$, for the next estimation process and the two-step prediction, $y'_{0,i+2,i}$, for the next detection process. The next detection process is, of course, carried out immediately after the receipt of r_{i+2}, and just before the next estimation

process.

The situation where $a_{0,i} = 0$ is handled as in Section 6.4.3, but, of course, now using the appropriate degree-1 least squares fading memory predictions in place of degree-0. The estimation and prediction of $y_{1,i}$ are determined in a manner similar to that for $y_{0,i}$.

D.2 Unbiased estimate of a very fast fading channel

The estimator here assumes that [1] ,

$$\begin{aligned}
&(y_{0,i+1} - y_{0,i}) - (y_{0,i} - y_{0,i-1}) \\
&\quad = (y_{0,i} - y_{0,i-1}) - (y_{0,i-1} - y_{0,i-2}) \\
&\quad \approx \dddot{y}_{0,i,i-1} \qquad \text{(D.20)}
\end{aligned}$$

Thus,

$$y_{0,i+1} - 2\,y_{0,i} + y_{0,i-1} = y_{0,i} - 2\,y_{0,i-1} + y_{0,i-2} \qquad \text{(D.21)}$$

$$\approx \dddot{y}_{0,i,i-1} \qquad \text{(D.22)}$$

so that,

$$3\,y_{0,i} = y_{0,i+1} + 3\,y_{0,i-1} - y_{0,i-2} \qquad \text{(D.23)}$$

and similarly for $y_{1,i}$. The estimator now operates on the four received samples, r_{i+1}, r_i, r_{i-1} and r_{i-2}, after having detected all the associaited data symbols, as before (see Section D.1). Correct detection of all data symbols is assumed in the following analysis.

The three received samples r_{i+1}, r_i and r_{i-1} are as given in eqns. D.3, D.4 and D.5, respectively, while the received sample, r_{i-2} , is given by

$$r_{i-2} = s_{0,i-2}\,y_{0,i-2} + s_{1,i-2}\,y_{1,i-2} + w_{i-2} \qquad \text{(D.24)}$$

From eqns. D.3, D.4, D.5 and D.24,

$$s_{1,i+1}^{-1} r_{i+1} = s_{1,i+1}^{-1} s_{0,i+1} y_{0,i+1} + y_{1,i+1} + s_{1,i+1}^{-1} w_{i+1} \quad \text{(D.25)}$$

$$3\, s_{1,i}^{-1} r_i = 3\, s_{1,i}^{-1} s_{0,i} y_{0,i} + 3\, y_{1,i} + 3\, s_{1,i}^{-1} w_i \quad \text{(D.26)}$$

$$3\, s_{1,i-1}^{-1} r_{i-1} = 3\, s_{1,i-1}^{-1} s_{0,i-1} y_{0,i-1} + 3\, y_{1,i-1} + 3\, s_{1,i-1}^{-1} w_{i-1} \quad \text{(D.27)}$$

$$s_{1,i-2}^{-1} r_{i-2} = s_{1,i-2.}^{-1} s_{0,i-2} y_{0,i-2} + y_{1,i-2} + s_{1,i-2}^{-1} w_{i-2} \quad \text{(D.28)}$$

From eqns. (D.25-D.28),

$$3\, s_{1,i}^{-1} r_i - s_{1,i+1}^{-1} r_{i+1} - 3\, s_{1,i-1}^{-1} r_{i-1} + s_{1,i-2}^{-1} r_{i-2}$$

$$= 3\, s_{1,i}^{-1} s_{0,i} y_{0,i} - s_{1,i+1}^{-1} s_{0,i+1} y_{0,i+1} - 3\, s_{1,i-1}^{-1} s_{0,i-1} y_{0,i-1}$$

$$+ s_{1,i-2}^{-1} s_{0,i-2} y_{0,i-2} + (\, 3\, y_{1,i} - y_{1,i+1} - 3\, y_{1,i-1} + y_{1,i-2}\,)$$

$$+ 3\, s_{1,i}^{-1} w_i - s_{1,i+1}^{-1} w_{i+1} - 3\, s_{1,i-1}^{-1} w_{i-1} + s_{1,i-2}^{-1} w_{i-2} \quad \text{(D.29)}$$

It now remains to eliminate $y_{1,i}$ from eqn. D.29. Toward this end, the estimator uses, from eqn. D.23, the relationship,

$$3\, y_{1,i} = y_{1,i+1} + 3\, y_{1,i-1} - y_{1,i-2} \quad \text{(D.30)}$$

and the fact that [2]

$$y_{0,i+1} \approx y_{0,i} + \dot{y}_{0,i,i-1} + \dddot{y}_{0,i,i-1} \quad \text{(D.31)}$$

$$y_{0,i-1} \approx y_{0,i} - \dot{y}_{0,i,i-1} + \dddot{y}_{0,i,i-1} \quad \text{(D.32)}$$

$$y_{0,i-2} \approx y_{0,i} - 2\,y''_{0,i,i-1} + 4\,y'''_{0,i,i-1} \qquad \text{(D.33)}$$

Substituting the values of eqns. (D.30-D.33) in eqn D.29, along with some manipulation of terms, gives,

$$\begin{aligned} & 3\,s^{-1}_{1,i}\,r_i - s^{-1}_{1,i+1}\,r_{i+1} - 3\,s^{-1}_{1,i-1}\,r_{i-1} + s^{-1}_{1,i-2}\,r_{i-2} \\ &= \Big(3\,s^{-1}_{1,i}\,s_{0,i} - s^{-1}_{1,i+1}\,s_{0,i+1} - 3\,s^{-1}_{1,i-1}\,s_{0,i-1} \\ &\quad + s^{-1}_{1,i-2}\,s_{0,i-2} \Big)\, y_{0,i} \\ &\quad - \Big(s^{-1}_{1,i+1}\,s_{0,i+1} - 3\,s^{-1}_{1,i-1}\,s_{0,i-1} + 2\,s^{-1}_{1,i-2}\,s_{0,i-2} \Big)\, y''_{0,i,i-1} \\ &\quad - \Big(s^{-1}_{1,i+1}\,s_{0,i+1} + 3\,s^{-1}_{1,i-1}\,s_{0,i-1} - 4\,s^{-1}_{1,i-2}\,s_{0,i-2} \Big)\, y'''_{0,i,i-1} \\ &\quad + \Big(3\,s^{-1}_{1,i}\,w_i - s^{-1}_{1,i+1}\,w_{i+1} - 3\,s^{-1}_{1,i-1}\,w_{i-1} \\ &\quad + s^{-1}_{1,i-2}\,w_{i-2} \Big) \end{aligned} \qquad \text{(D.34)}$$

Eqn. D.34 could now be expressed as

$$r_{0,i} \approx a_{0,i}\,y_{0,i} + u_{0,i} \qquad \text{(D.35)}$$

Eqn. D.35 is the same as eqn 6.4.20, albeit with its terms redefined as follows.

$$\begin{aligned} r_{0,i} &= 3\,s^{-1}_{1,i}\,r_i - s^{-1}_{1,i+1}\,r_{i+1} - 3\,s^{-1}_{1,i-1}\,r_{i-1} + s^{-1}_{1,i-2}\,r_{i-2} \\ &\quad + \Big(s^{-1}_{1,i+1}\,s_{0,i+1} - 3\,s^{-1}_{1,i-1}\,s_{0,i-1} + 2\,s^{-1}_{1,i-2}\,s_{0,i-2} \Big)\, y''_{0,i,i-1} \end{aligned}$$

$$+ \left(s^{-1}_{1,i+1} s_{0,i+1} + 3 s^{-1}_{1,i-1} s_{0,i-1} - 4 s^{-1}_{1,i-2} s_{0,i-2} \right) y'''_{0,i,i-1} \tag{D.36}$$

$$a_{0,i} = 3 s^{-1}_{i,1} s_{0,i} - s^{-1}_{1,i+1} s_{0,i+1} - 3 s^{-1}_{1,i-1} s_{0,i-1} + s^{-1}_{1,i-2} s_{0,i-2} \tag{D.37}$$

$$u_{0,i} = 3 s^{-1}_{1,i} w_i - s^{-1}_{1,i+1} w_{i+1} - 3 s^{-1}_{1,i-1} w_{i-1} + s^{-1}_{1,i-2} w_{i-2} \tag{D.38}$$

An unbiased estimate of $y_{0,i}$ is now given by $x_{0,i}$, where

$$x_{0,i} \approx a^{-1}_{0,i} r_{0,i} \approx y_{0,i} + a^{-1}_{0,i} u_{0,i} \tag{D.39}$$

with $r_{0,i}$, $a_{0,i}$ and $u_{0,i}$ given by eqns. (D.36 - D.38). The error in the one-step prediction $y'_{0,i,i-1}$, is then taken as (see eqn. 6.4.25),

$$\varepsilon_{0,i} = b \, |a_{0,i}| \, (x_{0,i} - y'_{0,i,i-1}) \tag{D.40}$$

where b is, as before, a small positive constant. Finally, the degree-2 least squares fading memory prediction algorithm (degree-2 polynomial filter in Table 6.4.1) is used to evaluate $y'_{0,i+1,i}$ and $y'_{0,i+2,i}$ for the next estimation and detection processes, respectively. The situation where $a_{i,0} = 0$ is handled as before, but with the appropriate degree-2 least squares fading memory predictions. The estimation and prediction of $y_{1,i}$ are determined in a manner similar to that for $y_{0,i}$.

References - Appendix D

1. **Clark, A.P. and McVerry, F.,** 'Channel estimators for an HF radio link', IEE Proc., Part F, Vol. 128, pp. 33-42, February 1981.

2. **Morrison, N.,** 'Introduction to sequential smoothing and prediction', McGraw Hill 1969.

APPENDIX E

THE EQUIVALENCE OF THE TESTED ESTIMATORS WHEN ONE SIGNAL IS RECEIVED

From eqn. 6.3.2, when one signal is received,

$$r_i = s_{0,i}\, y_{0,i} + w_i \qquad \text{(E.1)}$$

where the symbols r_i, $s_{0,i}$, $y_{0,i}$ and w_i are as defined in eqn. 6.3.2. Assume perfect detection.

Consider first the modified gradient estimator as described in Section 6.4.2. The error in the prediction $y'_{0,i,i-1}$ that is fed to the predictor is $\varepsilon_{0,i}$ (eqn . 6.4.9), where, with perfect dectection

$$\varepsilon_{0,i} = b\, e_i\, (s_{0,i})^* \qquad \text{(E.2)}$$

Substituting the values of eqns. 6.4.6, E.1 and 6.4.5, in eqn. E.2,

$$\begin{aligned} \varepsilon_{0,i} &= b\, (s_{0,i}\, y_{0,i} + w_i - s_{0,i}\, y'_{0,i,i-1})\, (s_{0,i})^* \\ &= b\, (s_{0,i})\, (y_{0,i} + s_{0,i}^{-1} w_i - y'_{0,i,i-1})\, (s_{0,i})^* \\ &= b\, |s_{0,i}|^2\, (y_{0,i} + s_{0,i}^{-1} w_i - y'_{0,i,i-1}) \end{aligned} \qquad \text{(E.3)}$$

Now consider the unbiased estimator as described in Section 6.4.3. When only one signal is received, the error in the prediction $y'_{0,i,i-1}$, that is fed to the predictor is $\varepsilon_{0,i}$, as given in eqn. 6.4.29, i.e.,

$$\varepsilon_{0,i} = b\, (s_{0,i}^{-1} r_i - y'_{0,i,i-1}) \qquad \text{(E.4)}$$

Substituting the value of eqn. 6.4.28 in eqn. E.4, the prediction error in the case of the unbiased estimator becomes

$$\varepsilon_{0,i} = b\,(y_{0,i} + s^{-1}_{0,i} w_i - y'_{0,i,i-1}) \qquad \text{(E.5)}$$

Now, consider eqns. E.3 and E.5, where the former gives the error that is fed to the predictor in the case of the modified gradient estimator, and the latter gives this same error in the case of the unbiased estimator. It could be seen that the quantity inside the main brackets is the same. Therefore, since b is a variable and since $| s_{0,i} |^2$ is a constant for a particular signal constellation, eqns .E.3 and E.5 could be adjusted to give exactly the same value for $\varepsilon_{0,i}$. As such, the prediction error that is fed to the least sqaures fading memory predictor, is the same for both the modified gradient estimator and the unbiased estimator, given that one signal is received. Thus, in the presence of one received signal, these two estimators are equivalent.

Q.E.D

APPENDIX F

ADAPTIVE ADJUSTMENT OF PREFILTER

In the ensuing discussion on the operation of the prefilter, it is assumed that the receiver has exact knowledge of the channel. As such, the prefilter is considered to operate on the sampled impulse response of the channel, rather than on an estimate (or prediction) of the sampled impulse response of the channel. The operation of the filter for a time invariant channel will be presented first, followed by an account of the modifications necessary to incorporate time varying channels.

Let the sampled impulse response of the channel be

$$Y = [\, y_0 \quad y_1 \quad ..... \quad y_g \,] \tag{F.1}$$

whose z-transform is

$$Y(z) = y_0 + y_1 z^{-1} + ... + y_g z^{-g} \tag{F.2}$$

Now, let

$$Y(z) = Y_1(z)\ Y_2(z) \tag{F.3}$$

where

$$Y_1(z) = \eta\ (1 + \alpha_1 z^{-1})(1 + \alpha_2 z^{-1}) ..... (1 + \alpha_{g-m} z^{-1}) \tag{F.4}$$

and

$$Y_2(z) = z^{-m}\ (1 + \beta_1 z)(1 + \beta_2 z) ..... (1 + \beta_m z) \tag{F.5}$$

with $|\alpha_i| < 1$ and $|\beta_i| < 1$ so that $Y_1(z)$ has all its roots inside the unit circle and $Y_2(z)$ has all its roots outside the unit circle. It is assumed that no roots of $Y(z)$ lie exactly on the unit circle. The quantity η is the complex value needed to satisfy eqns. (F.3 - F.5).

Let the adaptive prefilter have (n+1) taps. When the prefilter is ideally adjusted, the z-transform of its sampled impulse response is

$$D(z) = z^{-n}\, Y_2^{-1}(z)\; Y_3(z) \qquad \text{(F.6)}$$

where

$$Y_3(z) = (1+\beta_1^* z^{-1})(1+\beta_2^* z^{-1}) \;\ldots\ldots\; (1+\beta_m^* z^{-1}) \qquad \text{(F.7)}$$

and $(\beta_i)^*$ is the complex conjugate of β_i. From eqns. F.3 and F.6, it could be seen that the z-transform of the sampled impulse response of the cascade of the channel and prefilter is, approximately,

$$F(z) = Y(z)\; D(z)$$

$$= z^{-n}\, Y_1(z)\; Y_3(z) \qquad \text{(F.8)}$$

Eqn. F.8 is satisfied exactly when $n \Rightarrow \infty$. However, a good approximation may be obtained here without using an unduly large value of n [1,2]. It could be seen that the prefilter replaces those roots of Y(z) that lie outside the unit circle by the complex conjugates of their reciprocals, leaving the remaining roots unchanged, thereby making the response of the channel and prefilter, minimum phase. The algorithms for determining the wanted roots and for evaluating D(z) and F(z), arise from the following technique.

The receiver first forms a filter with the z-transform

$$A_i(z) = (1+\lambda_i z)^{-1} \qquad \text{(F.9)}$$

for i = 0, 1, ... k in turn, using an iterative process to adjust λ_i so that, as i increases, λ_i tends toward β_1, i.e. $\lambda_i \rightarrow \beta_1$. β_1 is the negative of the reciprocal of the first root to be processed by the system (eqn. F.5), and $|\beta_1| < 1$. Since the filter with z-transform $A_i(z)$ does not operate on the received signal in real time, its z-transform is not limited to zero and negative powers of z. At the end of the iterative process, the z-transform of the filter is

$$A_k(z) \approx (1+\beta_1 z)^{-1} \qquad \text{(F.10)}$$

The receiver next forms a filter with the z-transform

$$C_1(z) = (1+\lambda_k z)^{-1}(1+\lambda_k^* z^{-1})$$

$$\approx (1+\beta_1 z)^{-1}(1+\beta_1^* z^{-1}) \qquad \text{(F.11)}$$

The whole of this process is carried out for each β_h (h = 1, 2, ... m), to give a total of m filters with z-transforms $\{C_h(z)\}$. These m filters are then connected in cascade and a delay of (n-m) sampling intervals is added. The m filters and the associated delay are, in fact, implemented as a single filter whose z-transform now approximates D(z).

The algorithm for calculating D(z) and F(z) is as follows. The receiver first holds in store the sequence Y and an estimate λ_i, of the quantity β_1 . The first estimate of β_1 at the start of the process is one of a number of different starting points. The values of these starting points differ according to whether the channel is time invariant or otherwise, and as such, the appropriate start up procedure is given at the end of this Appendix, where the modifications necessary for time varying channels are considered.

Once λ_i has been determined, the receiver appropriately adjusts the one-tap feedback transversal filter shown in Fig. F.1. The stored sequence Y is now reversed in order, so that it starts with the component y_g , when it is fed through this filter. The sequence Y, passing through the filter in reverse order, is taken to be moving backwards in time, starting with the component y_0 at t = 0. The delay of one sampling interval, T, in the feedback filter, now becomes an advance of T with z-transform z. Thus the effective z-transform of the feedback filter becomes $A_i(z)$ (eqn. F.9) . The output from the filter is the sequence $\{e'_{i,h}\}$. Only the (g+1) components $e'_{i,0}$, $e'_{i,1}$, ... $e'_{i,g}$, of this sequence are, in fact, generated. An improved estimate of β_1 is now given by

$$\lambda_{i+1} = \lambda_i + \frac{c\, e'_{i,0}}{\varepsilon_i} \qquad \text{(F.12)}$$

where c is a constant in the range 0 to 1 and

$$\varepsilon_i = e'_{i,1} - e'_{i,2}\lambda_i + e'_{i,3}\lambda_i^2 - \ldots\ldots + e'_{i,g}(-\lambda_i)^{g-1} \qquad \text{(F.13)}$$

This gives a new one-tap feedback transversal filter with λ_i replaced by λ_{i+1} . The

effective z-transform of this filter, when operating on the sequence Y in reverse order, is

$$A_{i+1}(z) = (1 + \lambda_{i+1} z)^{-1} \qquad \text{(F.14)}$$

and the co-efficients of z^{-h} in $Y(z)\, A_{i+1}(z)$ is $e'_{i+1,h}$. The iterative process continues in this manner until one of the following occurs.

1. $$\left| \frac{e_{i,0}}{\varepsilon_i} \right| < d \qquad \text{(F.15)}$$

   where d is an appropriate small, positive, real constant.

2. $$i = 40 \qquad \text{(F.16)}$$

3. $$|\lambda_i| > 1 \qquad \text{(F.17)}$$

In each of the cases (1), (2), and (3), the process is terminated. In the case of (2) and (3), the iterative process is taken to have diverged, and the consequent actions for this will be described later. When condition (1) is met, the iterative process is taken to have converged. Let the value of i at convergence be k, such that,

$$\lambda_k \approx \beta_1 \qquad \text{(F.18)}$$

The receiver next appropriately adjusts the two-tap feedforward transversal filter shown in Fig. F.2, which has the z-transform

$$B_k(z) = 1 + \lambda_k^* z^{-1} \qquad \text{(F.19)}$$

The sequence $\{e'_{k,h}\}$ is fed through this two-tap feedforward transversal filter in the correct order, to give the (g+2)-component output sequence

$$f_{1,-1} + f_{1,0} z^{-1} + \ldots\ldots\ldots + f_{1,g} z^{-(g+1)} \qquad \text{(F.20)}$$

which is approximately equal to $Y(z)\ A_k(z)\ B_k(z)$, and where $f_{1,-1} \approx 0$. The resulting effect of the sequence Y passing through the two filters in Figs. F.1 and F.2, is the same as that of the sequence Y passing through a single filter with the z-transform

$$C_1(z) = A_k(z)\ B_k(z) \tag{F.21}$$

From eqns. F.14, F.18, F.19 and F.21,

$$C_1(z) \approx (1+\beta_1 z)^{-1}\ (1+\beta_1^* z^{-1}) \tag{F.22}$$

as in eqn. F.11. Finally, the output sequence, $\{f_{1,h}\}$, is advanced by one sampling interval (multiplication by z), and the first component, $f_{1,-1}$, is discarded, to give the sequence F_1, with the z-transform

$$F_1(z) = f_{1,0} + f_{1,1} z^{-1} + ... + f_{1,g} z^{-g} \tag{F.23}$$

$$\approx z\ Y(z)\ C_1(z) \tag{F.24}$$

For practical purposes, the linear factor $(1+\beta_1 z)$ in eqn. F.5, is replaced in $F_1(z)$ by the linear factor $(1+\beta_1^* z^{-1})$. Thus, the root $-1/\beta_1$, of Y(z), is replaced by the root $(-\beta_1)^*$, i.e., one root of Y(z) which lay outside the unit circle is replaced by the complex conjugate of its reciprocal, and hence now lies inside the unit circle. $F_1(z)$ contains, in addition, an advance of one sampling interval. The iterative process is now repeated with Y replaced by F_1, for the tracking of more roots.

The adjustment of the tap gains of the filter whose ideal z-transform is D(z), (eqn. F.6), will now be explained. All the tap gains of this (n+1)-tap linear feedforward transversal filter are initially set to zero, except for the last tap, whose gain is set to unity. Thus, the initial z-transform of the filter is

$$D_0(z) = z^{-n} \tag{F.25}$$

and the initial z-transform of the channel and filter is $z^{-n}\ Y(z)$. When convergence has been

obtained in the iterative process previously described, such that $\lambda_k \approx \beta_1$, the sequence D_0 is fed through the two-tap feedforward transversal filter with z-transform $B_k(z)$ (Fig. F.2), starting with the first component of D_0 . This gives an output sequence with (n+2) components and z-transform $D_0(z)\, B_k(z)$. This output sequence is now fed in reverse order (starting with the last component) through the one-tap feedback transversal filter shown in Fig. F.1. The effective z-transform of the one-tap feedback transversal filter is now $A_k(z)$, and hence the output sequence from this filter will have a z-transform that is approximately

$$D_0(z)\; A_k(z)\; B_k(z) \;=\; D_0(z)\; C_1(z) \qquad \text{(F.26)}$$

When (n+1) components of the output sequence have been obtained, the process is halted. These (n+1) components, in the order in which they are received, are the co-efficients of $z^{-(n+1)}$, z^{-n} , ... , z^{-1} , in $D_0(z)\, C_1(z)$. The tap gains of the h th tap of the adaptive filter is now set to the co-efficient z^{-h} (for $h = 1, 2, \ldots\ldots, n+1$), to give the required tap gains. Thus, the z-transform of the adaptive filter is, approximately,

$$D_1(z) \;=\; z\; D_0(z)\; C_1(z) \qquad \text{(F.27)}$$

The whole of the root finding process and tap adjustment process just described, is now repeated, albeit using $F_1(z)$ in eqn. F.12 (in place of Y(z)), and $D_1(z)$ in eqn. F.25 (in place of $D_0(z)$). At the end of the root finding iterative process, $\lambda_k \approx \beta_2$, and as such, the values of $F_1(z)$, λ_k , and $D_1(z)$ determine the values of $F_2(z)$ and $D_2(z)$. $F_2(z)$ and $D_2(z)$ are now used in place of $F_1(z)$ and $D_1(z)$, in order to process β_3 .

The whole process continues this way until no roots of $F_h(z)$ outside the unit circle are found, starting from any possible λ_1 . It is assumed that no roots of $F_h(z)$ lie outside the unit circle when one of the two divergence conditions (eqn. F.16 and F.17) are met, regardless of the starting point λ_1 . Thus, it can now be assumed that all the m roots of Y(z) that lie outside the unit circle have been replaced by the complex conjugate of their reciprocals, in the z-transform of the channel and adaptive filter. The z-transform, now, of the adaptive filter is

$$D_m(z) \;\approx\; D(z) \qquad \text{(F.28)}$$

so that the z-transform of the channel and adaptive filter is

$$Y(z)\ D_m(z) \approx a_0 + a_1 z^{-1} + \ldots\ldots\ldots + a_{n+g} z^{-(n+g)} \qquad \text{(F.29)}$$

where $a_h \approx 0$ for $h = 0, 1, \ldots\ldots, (n-1)$.

The estimate of the sampled impulse response of the channel and adaptive filter, that is employed by the detector, is the sequence F_m with z-transform

$$F_m(z) = f_{m,0} + f_{m,1} z^{-1} + \ldots\ldots\ldots + f_{m,g} z^{-g} \qquad \text{(F.30)}$$

$$\approx z^n F(z) \qquad \text{(F.31)}$$

Thus, from eqn. F.8,

$$F_m(z) = Y_1(z)\ Y_3(z) \qquad \text{(F.32)}$$

The delay of n sampling intervals introduced by the adaptive filter is, for convenience, ignored here, but should be taken into account when comparing $Y(z)\ D_m(z)$ and $F_m(z)$.

The above are the algorithms for the estimation of the sampled impulse response of the channel and filter, and for the adjustment of the tap gains of the filter, when the channel is considered to be time invariant. For time varying channels, the following modifications should be implemented.

Firstly, the above algorithms should be implemented at every sampling instant, such that both F_m and the tap gains are calculated at each t=iT, for i = 1, 2, ... If the channel is varying only very slowly with time, however, it may be sufficient to perform the adjustment of the filter tap gains once every so often, say, once every four sampling instants. The work in this thesis is based on the adaptation of the filter at every sampling instant.

Secondly, nine starting points are used for the adaptive filter, when operating over time varying channels. The values of these starting points are given later . The algorithm always starts with λ_1 set to starting point number 1 (see later), and it

uses all the nine possible starting points for tracking any new roots. This is done in order to facilitate the tracking of as many as possible of the roots outside the unit circle, at the beginning. When the algorithm diverges (eqns. F.16 and F.17) for each of the nine starting points, it is assumed that all of the m roots that lie outside the unit circle have been found. For subsequent runs, however, the m roots that were tracked in the previous run, are added to the nine starting points, to give a total of (m+9) starting points. These (m+9) starting points are arranged as shown in Fig. F.3, and the algorithm now starts with λ_1 set to β_1, where β_1 here is the first root tracked in the previous run. Whenever a root is found or a new starting point is required due to divergence of the algorithm, the next of the (m+9) starting points is used. This process is repeated until all (m+9) starting points have been used, at which stage it is assumed that there are no more roots of Y(z) that lie outside the unit circle, and the process is terminated. Thus, once a set of roots is found, it is added to the original nine starting points, such that the starting points for any subsequent run are the roots tracked from the previous run, plus the original nine starting points.

Another possible modification for the operation of the adaptive filter over time varying channels is to redefine eqn. F.16 as

$$i = 100 \qquad \text{(F.33)}$$

so that greater latitude is allowed in the iteration process, before determining that the process has diverged. This modification is only necessary, however, in the very worst of situations. In general, tests have shown that eqn. F.16 is a reasonable condition for divergence, and the work in this thesis assumes this fact.

The nine starting points used in the adaptive filter, when operating over time varying channels, are,

1. 0.00000 + j 0.00000
2. 0.90909 + j 0.00000
3. 0.00000 - j 0.90909
4. 0.00000 + j 0.90909
5. -0.90909 + j 0.00000
6. 0.64282 - j 0.64282
7. 0.64282 + j 0.64282
8. -0.64282 + j 0.64282
9. -0.64282 - j 0.64282

References - Appendix F

1. **Clark, A.P. and Hau, S.F.,** ' Adaptive adjustment of receiver for distorted digital signals ' , IEE Proc. Part F, August, 1984.

2. **Abdullah, S.N.,** ' Detection processes for a 9600 kbit/s voice modem ', Ph.D. thesis, Loughborough University of Technology, 1986.

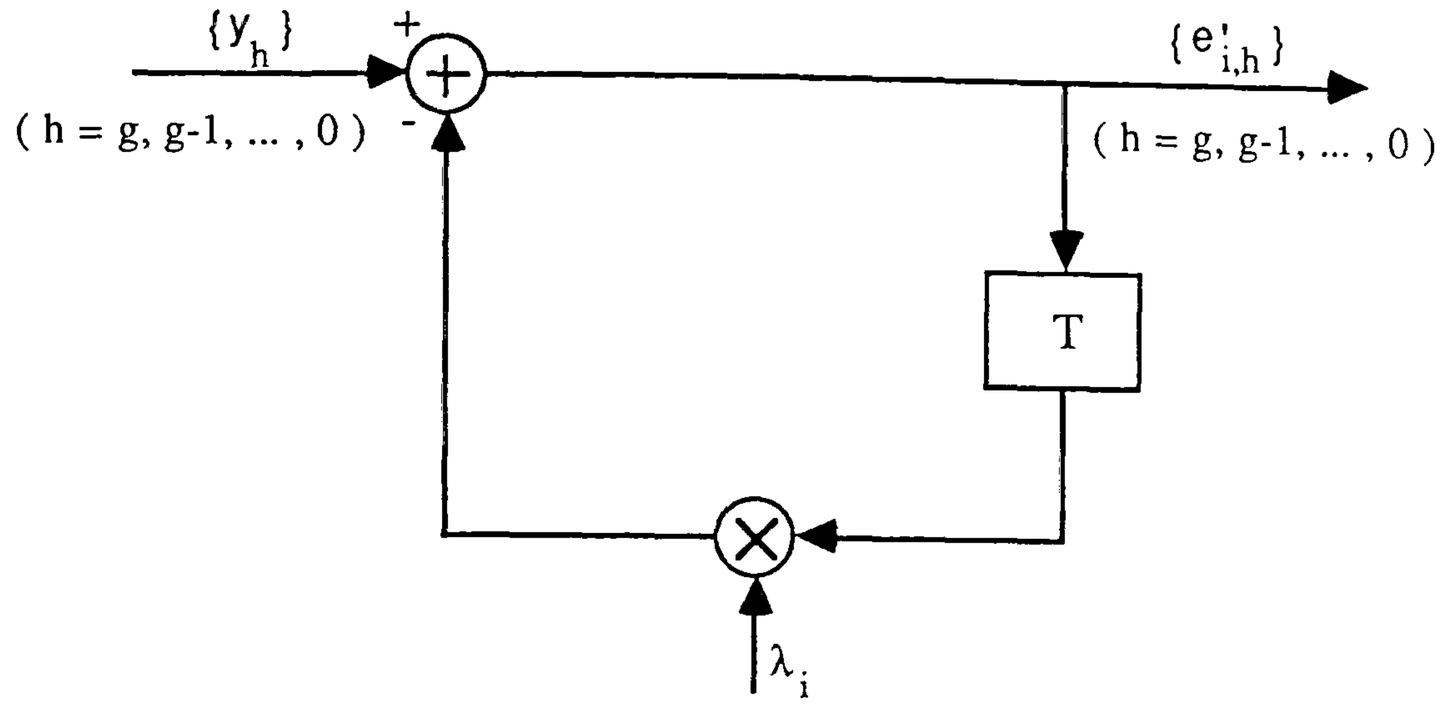


Fig. F.1 One-tap feedback filter

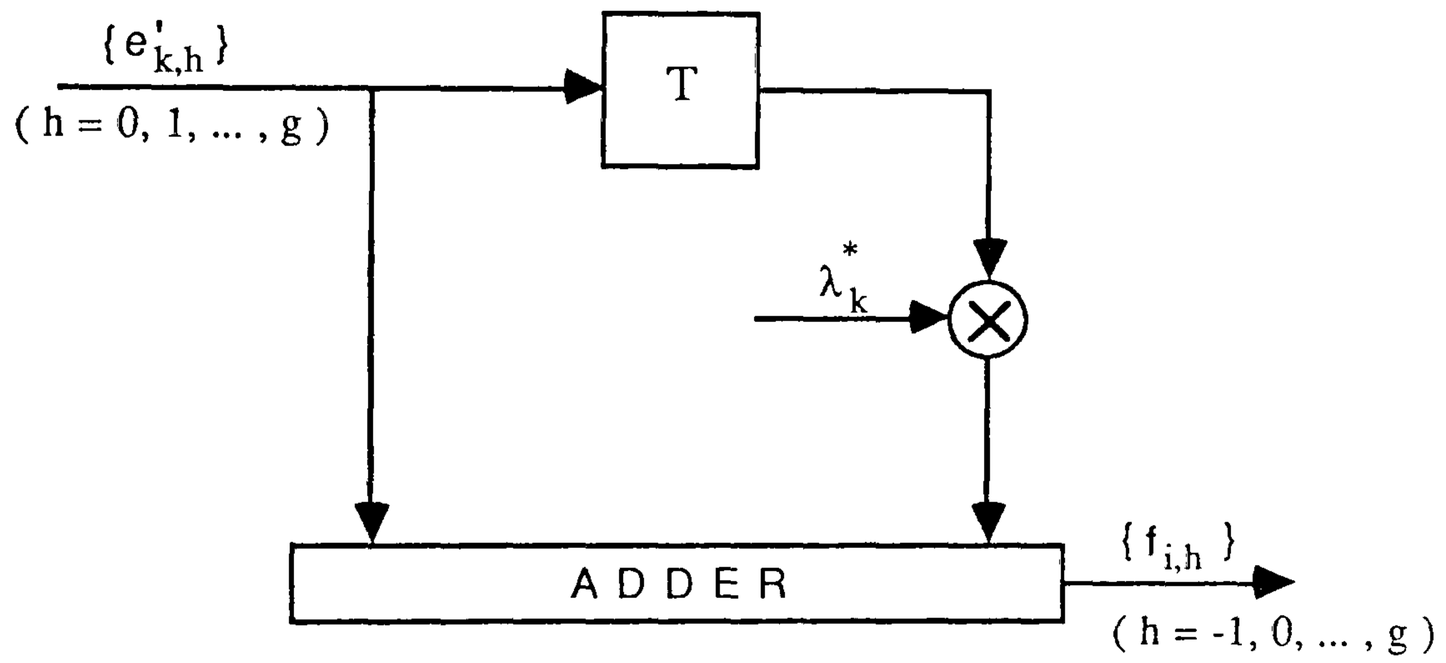


Fig. F.2 Two-tap feedforward transversal filter

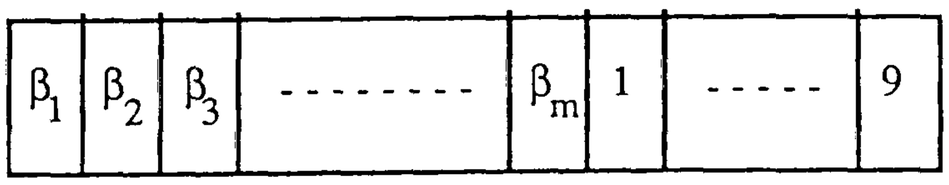


Fig. F.3 Arrangement of starting points for the algorithm

APPENDIX G

SIMULATION OF 3-SKYWAVE HF RADIO CHANNEL INCORPORATING IDEAL MINIMUM PHASING

```
/*JOB FZ36C,EUELSJ2,ST=C20,C=D,TI=2500,
/* PW=SDM30
PATTACH,LIBAPPL.
PATTACH,PROCLIB.
FTN200,L=0,OPT.
LOAD,LIB=NAGLIB.
GO.
BEGIN,,PUTFEP,FDIMP,JC36,FO=BINARY.
####S
      PROGRAM FADE
      PARAMETER (KPAR1=16,KPAR2=30,KPAR3=22,LLOUT=5)
      REAL CF(5),Q(6,2600)
      REAL TXR(KPAR1),TXI(KPAR1),TXDR(KPAR1),TXDI(KPAR1)
      REAL TXDDR(KPAR1),TXDDI(KPAR1)
      REAL RXR(KPAR2),RXI(KPAR2),WSR(KPAR2),WSI(KPAR2)
      REAL QR(KPAR2,KPAR2),QI(KPAR2,KPAR2)
      REAL YR(KPAR2),YI(KPAR2)
      REAL CON(6),QQ(6),EQ(6),VQ(6),FMEAN(6),FVAR(6)
      REAL YNAGR(KPAR3),YNAGI(KPAR3),REZ(KPAR3),IMZ(KPAR3)
      REAL YCOFR(KPAR3),YCOFI(KPAR3),DUMR(KPAR3),DUMI(KPAR3)
      REAL YNDIVR(KPAR3),YNDIVI(KPAR3)

C     VALUES TO VARIABLES AND ARRAYS

      DATA CF/-0.1803229723D+01,0.8152066804D+00,-0.1852182882D+01,
     1        0.8678845458D+00,-0.8948130729D+00/
      DCG=19378
      DCG=1.0/DCG
      STDVN=SQRT(1.0)
      NOSAM=2500
      INFD=36
      FSP=2.0


C     INITIALISING

      DL1=0.0
      DL2=0.0
      DL3=0.0
      DL4=0.0
      DL5=0.0

      JQ=50+NOSAM
      JQ1=JQ+1


      OPEN(LLOUT,FILE='OUTPUT',IOSTAT=JOSS)
      OPEN(8,FILE='FDIMP',FORM='UNFORMATTED',IOSTAT=JJOSS)
      WRITE(LLOUT,150)JOSS
      WRITE(LLOUT,160)JJOSS
150   FORMAT(/' ERRORS IN EXECUTION OF 1ST OPEN STATEMENT=',I3)
160   FORMAT(' ERRORS IN EXECUTION OF 2ND OPEN STATEMENT=',I3)

      CALL G05CBF(INFD)

C     GENERATION OF Q1(T) AND Q2(T)
```

```
      DO 250 I=1,6

      JA=1

      TF3=0.0
      TVF3=0.0
      TF3DCG=0.0
      TVF3DCG=0.0


          DO 240 J=1,JQ1

          F0=G05DDF(0.00D00,STDVN)
          F1=F0-(DL1*CF(1)+DL2*CF(2))
          F2=F1-(DL3*CF(3)+DL4*CF(4))
          F3=F2-(DL5*CF(5))

          F3DCG=F3*DCG

          DL5=F3
          DL4=DL3
          DL3=F2
          DL2=DL1
          DL1=F1

          IF(J.LE.50) GO TO 240

          Q(I,JA)=F3DCG
          JA=JA+1


          TF3=TF3+F3
          TVF3=TVF3+(F3**2)
          TF3DCG=TF3DCG+F3DCG
          TVF3DCG=TVF3DCG+(F3DCG**2)


      240       CONTINUE


          EF3=TF3/(NOSAM+1)
          VARF3=TVF3/(NOSAM+1)
          EF3DCG=TF3DCG/(NOSAM+1)
          VARF3DC=TVF3DCG/(NOSAM+1)


          WRITE(LLOUT,244)EF3,VARF3
          WRITE(LLOUT,246)EF3DCG,VARF3DC
      244   FORMAT(' MEAN OF F3 =',1X,E20.10,2X,' VAR. OF F3 =',1X,E20.10)
      246   FORMAT(' MEAN OF F3DCG =',1X,E20.10,2X,' VAR. OF F3DCG =',1X,
         1 E20.10)

      250   CONTINUE


      C     INITIALISING ARRAYS AND VARIABLES FOR MAIN PROGRAM

          DATA TXR /-0.1795896, -3.0773455, -9.9409021,-11.7869473,
         1        -3.4618271,  4.4438154,  3.0642536, -1.3596576,
         2        -1.4973528,  0.2925598,  0.5180829, -0.1842786,
         3        -0.3167778,  0.0021899, -0.0443806,  0.0515533 /

          DATA TXI / 2.3539405, 20.7590237, 45.5584592, 41.4909978,
         1        8.7045826,-11.7869820, -5.5819054,  3.1582131,
```

```
2         1.7365460, -0.7776891, -0.1292556,  0.2880296,
3        -0.2324818, -0.2107548,  0.0392056,  0.0098505 /

 DATA TXDR / -1.6694374, -7.8492148,-12.3887079, -6.6023157,
1          2.9408554,  4.3005084, -0.3368383, -1.9014342,
2         -0.1433592,  0.6242601,  0.0278577, -0.3820071,
3         -0.0416905, -0.0439705,  0.0749333, -0.0594132 /

 DATA TXDI / 13.2372707, 39.6493461, 46.9272219, 19.2346609,
1         -8.8804125, -9.0256163,  1.6284281,  2.8139013,
2         -0.4311352, -0.4537174,  0.3081762, -0.0772327,
3         -0.3043271,  0.0085057,  0.0093809,  0.0094992 /

 DATA TXDDR / -1.3136537, -7.1104051,-12.3469721, -7.5848703,
1          2.2353854,  4.5938614,  0.0931639, -1.9704176,
2         -0.3233694,  0.6313238,  0.1035718, -0.3865939,
3         -0.0734526, -0.0386471,  0.0608046, -0.0713496 /

 DATA TXDDI / 11.0688962, 37.2136597, 47.9575159, 22.8262482,
1         -7.2498590,-10.0026703,  0.8695437,  3.1072800,
2         -0.2261096, -0.5552906,  0.2882096, -0.0156703,
3         -0.3215770, -0.0107706,  0.0140909,  0.0135711 /

 DATA RXR / -1.9417691,-15.9797864,-35.1417733,-34.4788717,
1        -11.2301982,  7.8155160,  7.5124057, -0.5057505,
2         -3.3707125, -0.6759166,  1.0482656,  0.3621876,
3         -0.3105902,  0.0438410,  0.0738947, -0.0646936,
4          0.0000000,  0.0000000,  0.0000000,  0.0000000,
5          0.0000000,  0.0000000,  0.0000000,  0.0000000,
6          0.0000000,  0.0000000,  0.0000000,  0.0000000,
7          0.0000000,  0.0000000 /

 DATA RXI /  1.3625952, 11.5941040, 27.3342937, 28.0870086,
1          7.2714615, -9.2602472, -5.0954462,  3.2326498,
2          1.8975352, -1.2813604, -0.4830313,  0.7614804,
3          0.1979014, -0.1532672,  0.0940330, -0.0312132,
4          0.0000000,  0.0000000,  0.0000000,  0.0000000,
5          0.0000000,  0.0000000,  0.0000000,  0.0000000,
6          0.0000000,  0.0000000,  0.0000000,  0.0000000,
7          0.0000000,  0.0000000 /


 NLOOP=1200
 MLOOP=2*NLOOP
 ISTEP=48
 STEP=1.0/ISTEP
 DEL1=1.1
 DEL2=3.0
 SAMPRAT=2.4
 SFACT=1.0/(SAMPRAT*2*1000.0)
 IDEL1=INT(SAMPRAT*2*DEL1)
 IDEL2=INT(SAMPRAT*2*DEL2)
 KMPL=16
 KMP=IDEL2+KMPL
 KMP1=KMP-1
 KMCONV=(KMP+KMPL-1)/2
 UCIRC=1.05
 INDTN=15
 POS=-1.0
 SQ=0.0
 SSQ=0.0
 JCOUNT=0
 ICOUNT=0
```

```
      DO 1010 I=1,KMP
      DO 1005 J=1,KMP
      QR(I,J)=0.0
      QI(I,J)=0.0
1005  CONTINUE
1010  CONTINUE

      DO 1020 I=1,6
      EQ(I)=0.0
      VQ(I)=0.0
1020  CONTINUE

      DO 1030 I=1,KMCONV
      REZ(I)=0.0
      IMZ(I)=0.0
1030  CONTINUE

      CALL G05CBF(INDTN)


C     ENTERING MAIN LOOP

      DO 9000 KMAIN=1,MLOOP


      DO 1510 I=1,6
      CON(I)=(Q(I,KMAIN+1)-Q(I,KMAIN))*STEP
1510  CONTINUE


C     ENTERING SECONDARY LOOP
      DO 8000 KSEC=1,ISTEP


      ICOUNT=ICOUNT+1
      COUNT=REAL(ICOUNT)

      DO 1520 I=1,6
      QQ(I)=Q(I,KMAIN)+((KSEC-1)*CON(I))
1520  CONTINUE

      QQ(2)=-QQ(2)
      QQ(4)=-QQ(4)
      QQ(6)=-QQ(6)

      DO 1530 I=1,6
      EQ(I)=EQ(I)+QQ(I)
      VQ(I)=VQ(I)+(QQ(I)*QQ(I))
1530  CONTINUE


C     SHIFTING ARRAYS FOR CONVOLUTION

      DO 2010 I=1,KMP
      DO 2005 J=1,KMP1
      QR(I,KMP+1-J)=QR(I,KMP-J)
      QI(I,KMP+1-J)=QI(I,KMP-J)
2005  CONTINUE
2010  CONTINUE
      DO 2015 I=1,KMP
      QR(I,1)=0.0
      QI(I,1)=0.0
2015  CONTINUE
```

```
C     CONVOLUTION (TO OBTAIN IMPULSE RESPONSE OF CHANNEL), BEGINS

      DO 2020 I=1,KMPL
      QR(I,1)=TXR(I)*QQ(1)-TXI(I)*QQ(2)
      QI(I,1)=TXR(I)*QQ(2)+TXI(I)*QQ(1)
2020  CONTINUE
      DO 2030 I=1,KMPL
      QR(I+IDEL1,1)=QR(I+IDEL1,1)+TXDR(I)*QQ(3)-TXDI(I)*QQ(4)
      QI(I+IDEL1,1)=QI(I+IDEL1,1)+TXDR(I)*QQ(4)+TXDI(I)*QQ(3)
2030  CONTINUE
      DO 2040 I=1,KMPL
      QR(I+IDEL2,1)=QR(I+IDEL2,1)+TXDDR(I)*QQ(5)-TXDDI(I)*QQ(6)
      QI(I+IDEL2,1)=QI(I+IDEL2,1)+TXDDR(I)*QQ(6)+TXDDI(I)*QQ(5)
2040  CONTINUE

      POS=-POS
      IF(POS.LT.0) GO TO 8000


      IO=0
      JCOUNT=JCOUNT+1
      DCOUNT=REAL(JCOUNT)

                          5
      DO 2060 I=1,KMP,2
      IO=IO+1
      YR(IO)=0.0
      YI(IO)=0.0
       DO 2050 J=1,I
       YR(IO)=YR(IO)+QR(J,I+1-J)*RXR(I+1-J)-QI(J,I+1-J)*RXI(I+1-J)
        YI(IO)=YI(IO)+QI(J,I+1-J)*RXR(I+1-J)+QR(J,I+1-J)*RXI(I+1-J)
2050    CONTINUE
      YR(IO)=YR(IO)*SFACT
      YI(IO)=YI(IO)*SFACT
2060  CONTINUE

      IF (MOD(KMP,2).EQ.0) THEN
      GO TO 2070
      ELSE
      GO TO 2100
      END IF

2070  DO 2090 I=1,KMP1,2
      IO=IO+1
      YR(IO)=0.0
      YI(IO)=0.0
      MCONV=I+1
       DO 2080 J=MCONV,KMP
       KCVT=KMP+1+I-J
       YR(IO)=YR(IO)+QR(J,KCVT)*RXR(KCVT)-QI(J,KCVT)*RXI(KCVT)
       YI(IO)=YI(IO)+QI(J,KCVT)*RXR(KCVT)+QR(J,KCVT)*RXI(KCVT)
2080    CONTINUE
      YR(IO)=YR(IO)*SFACT
      YI(IO)=YI(IO)*SFACT
2090  CONTINUE

      GO TO 2150

2100  DO 2120 I=1,KMP1,2
      IO=IO+1
      YR(IO)=0.0
      YI(IO)=0.0
```

```
      MCONV=I+2
       DO 2110 J=MCONV,KMP
       YR(IO)=YR(IO)+QR(J,KMP+2+I-J)*RXR(KMP+2+I-J)
     1          -QI(J,KMP+2+I-J)*RXI(KMP+2+I-J)
       YI(IO)=YI(IO)+QI(J,KMP+2+I-J)*RXR(KMP+2+I-J)
     1          +QR(J,KMP+2+I-J)*RXI(KMP+2+I-J)
2110    CONTINUE
      YR(IO)=YR(IO)*SFACT
      YI(IO)=YI(IO)*SFACT
2120  CONTINUE

2150  CONTINUE
```

```
C     OBTAINING ROOTS OF Z-TRANSFORM OF CHANNEL

      DO 3000 I=1,KMCONV
      YNAGR(I)=YR(I)
      YNAGI(I)=YI(I)
3000  CONTINUE
      NROOT=KMCONV
      TOL=X02AAF(1.0)
      IFAIL=0
      CALL C02ADF(YNAGR,YNAGI,NROOT,REZ,IMZ,TOL,IFAIL)


C     MANIPULATION OF ROOTS TO MAKE CHANNEL MINIMUM PHASE

      RPROD=1.0
      DO 3010 I=1,KMCONV
      RSQMOD=REZ(I)*REZ(I)+IMZ(I)*IMZ(I)
      RMOD=RSQMOD**0.5
      IF(RMOD.GT.UCIRC)THEN
      RPROD=RPROD*RMOD
      REZ(I)=REZ(I)/RSQMOD
      IMZ(I)=IMZ(I)/RSQMOD
      ELSE
      GO TO 3010
      END IF
3010  CONTINUE


C     RECONSTRUCTION OF CHANNEL

      DO 3020 I=1,KMCONV
      YCOFR(I)=0.0
      YCOFI(I)=0.0
3020  CONTINUE
      YCOFR(1)=1.0
```

```
      DO 3050 I=1,KMCONV-1
         DO 3030 J=1,KMCONV-1
         DUMR(J)=YCOFR(J)*REZ(I)-YCOFI(J)*IMZ(I)
         DUMI(J)=YCOFR(J)*IMZ(I)+YCOFI(J)*REZ(I)
3030       CONTINUE
         DO 3040 JJ=1,KMCONV-1
         JJ1=JJ+1
         YCOFR(JJ1)=YCOFR(JJ1)+DUMR(JJ)
         YCOFI(JJ1)=YCOFI(JJ1)+DUMI(JJ)
3040       CONTINUE
3050  CONTINUE


      DO 3060 I=2,KMCONV,2
```

```
      YCOFR(I)=-YCOFR(I)
      YCOFI(I)=-YCOFI(I)
 3060 CONTINUE

      DO 3070 I=1,KMCONV
      YNDIVR(I)=(YCOFR(I)*YR(1)-YCOFI(I)*YI(1))*RPROD
      YNDIVI(I)=(YCOFR(I)*YI(1)+YCOFI(I)*YR(1))*RPROD
 3070 CONTINUE

      WRITE(8)(YNDIVR(I),I=1,KMCONV)
      WRITE(8)(YNDIVI(I),I=1,KMCONV)


C     CHECKING THE SUM OF THE SQUARES

      YSQ=0.0
      YYSQ=0.0
      DO 3090 I=1,KMCONV
      YSQ=YSQ+(YR(I)*YR(I)+YI(I)*YI(I))
      YYSQ=YYSQ+(YNDIVR(I)*YNDIVR(I)+YNDIVI(I)*YNDIVI(I))
 3090 CONTINUE
      SQ=SQ+YSQ
      SSQ=SSQ+YYSQ
      YDBQ=10.0*LOG10(YYSQ)

      IF (JCOUNT.EQ.1.OR.MOD(JCOUNT,10000).EQ.0) THEN
      WRITE(LLOUT,4500)(YNDIVR(I),I=1,KMCONV)
      WRITE(LLOUT,4500)(YNDIVI(I),I=1,KMCONV)
 4500 FORMAT(4E15.5)
      END IF


 8000 CONTINUE


 9000 CONTINUE


      DO 9100 I=1,6
      FMEAN(I)=EQ(I)/COUNT
      FVAR(I)=VQ(I)/COUNT
 9100 CONTINUE

      AVSQ=SQ/DCOUNT
      AVSSQ=SSQ/DCOUNT


C     PRINTING RESULTS

      WRITE(LLOUT,9203)FSP,DEL1,DEL2,SAMPRAT
 9203 FORMAT(' FREQUENCY SPREAD                =',F6.2/
     1      ' DELAY BETWEEN SKYWAVE NOS. 1 AND 2 =',F6.3/
     2      ' DELAY BETWEEN SKYWAVE NOS. 2 AND 3 =',F6.3/
     3      ' SAMPLING RATE IN KILOBAUDS        =',F6.3)

      WRITE(LLOUT,9205)INDTN,ICOUNT,JCOUNT,KMCONV
 9205 FORMAT(' INDTN=',I7/' ICOUNT=',I7/' JCOUNT=',I7/' KMCONV=',I4)

      WRITE(LLOUT,9210)
 9210 FORMAT(' MEAN VALUE OF THE QQ-SIGNALS')
      WRITE(LLOUT,9230)(FMEAN(I),I=1,6)
      WRITE(LLOUT,9220)
 9220 FORMAT(/' VARIANCES OF THE QQ-SIGNALS')
      WRITE(LLOUT,9230)(FVAR(I),I=1,6)
```

```
9230  FORMAT(E20.10)

      DO 9235 I=1,KMCONV
      WRITE(LLOUT,9240)YR(I),YI(I),YNDIVR(I),YNDIVI(I)
9235  CONTINUE
9240  FORMAT(F10.5,3X,F10.5,3X,F10.5,3X,F10.5)

      WRITE(LLOUT,9250)AVSQ
9250  FORMAT(' AVERAGE SUM OF SQUARES OF ORIGINAL CHANNEL=',F7.4)
      WRITE(LLOUT,9260)AVSSQ
9260  FORMAT(' AVERAGE SUM OF SQUARES OF MIN.PHASE CHANNEL=',F7.4///)

      STOP
      END
####S
```

APPENDIX H

SIMULATION OF TRELLIS CODED MODULATION SYSTEM D16

```
/*JOB DQRF1,EUELSJ2,ST=C20,C=C,TI=2500,
/* PW=SDMVP
PATTACH,LIBAPPL.
PATTACH,PROCLIB.
BEGIN,,GETFEP,FDIMP,JC36,FO=BINARY.
SWITCH,FDIMP,RT=W.
FTN200,L=0,OPT.
LOAD,LIB=BLAS,NAGLIB.
GO.
####S
      PROGRAM DQRF
      PARAMETER(NDEL=32,PI=3.1415926536,NIMP=22,NVEC=16,NSTAT=8)
      PARAMETER(KLOUT=99)
      INTEGER IBIT1(NDEL),IBIT2(NDEL),IDBIT1(NDEL),IDBIT2(NDEL)
      INTEGER ISTE(NDEL),IVOUT(NDEL)
      INTEGER ILP1(32),ILKUP1(16,2),ILP4(32),ILKUP4(16,2)
      INTEGER ILP2(32),ILP3(32),ILKUP2(0:7,0:3),ILKUP3(0:7,0:3)
      INTEGER ILP5(64),ILKUP5(8,8)
      INTEGER LST1(NSTAT,NVEC/NSTAT)
      INTEGER ITX1(NVEC,NDEL),ITX2(NVEC,NDEL)
      INTEGER JTX1(NVEC*4,NDEL),JTX2(NVEC*4,NDEL)
      INTEGER LMTR(NVEC*4),LMTI(NVEC*4)
      REAL SR(NDEL),SI(NDEL)
      REAL VSR(NVEC,NDEL),VSI(NVEC,NDEL)
      REAL YR(NIMP),YI(NIMP)
      REAL YTR(NIMP),YTI(NIMP)
      REAL HFR(NIMP),HFI(NIMP),WFR(NIMP),WFI(NIMP)
      REAL GR(NVEC),GI(NVEC)
      REAL TR(NVEC*4,NDEL),TI(NVEC*4,NDEL)
      REAL PLIUP1(8,4),PLIUP2(8,4)
      REAL COST(NVEC),CT1(NVEC*4),CT2(NSTAT,NVEC/NSTAT)

C     VALUES TO MAIN VARIABLES AND ARRAYS

      N=NDEL
      M=NVEC
      MM=NSTAT
      IQ=27
      KLOOP=57600
      NTR=KLOOP/10
      NOSNR=5
      SNR=7.5
      SNRD=0.5
      SNR=SNR-SNRD
      N1=N-1
      M4=M*4
      MDM=M/MM
      ML1=MDM*4
      ROOT2=SQRT(2.0)


      DATA WFR/-0.02805,-0.23081,-0.50758,-0.49800,
     1     -0.16221, 0.11289, 0.10851,-0.00731,
     2     -0.04869,-0.00976, 0.01514, 0.00523,
     3     -0.00449, 0.00063, 0.00107,-0.00093,
     4      0.00000, 0.00000, 0.00000, 0.00000,
```

```
     5       0.00000, 0.00000/

      DATA WFI/ 0.01968, 0.16746, 0.39481, 0.40568,
     1       0.10503,-0.13375,-0.07360, 0.04669,
     2       0.02741,-0.01851,-0.00698, 0.01100,
     3       0.00286,-0.00221, 0.00136,-0.00045,
     4       0.00000, 0.00000, 0.00000, 0.00000,
     5       0.00000, 0.00000/


      DATA ILP1/0,0,0,1,1,0,1,1,0,1,1,0,1,1,0,0,
     1       1,0,1,1,0,0,0,1,1,1,0,0,0,1,1,0/

      DATA ILP2/0,1,0,1,2,3,2,3,4,5,4,5,6,7,6,7,
     1       0,1,0,1,2,3,2,3,4,5,4,5,6,7,6,7/

      DATA ILP3/0,2,4,6,1,3,5,7,3,1,7,5,2,0,6,4,
     1       6,4,2,0,7,5,3,1,1,3,5,7,0,2,4,6/

      DATA ILP4/0,0,0,1,1,0,1,1,1,1,0,0,0,1,1,0,
     1       1,0,1,1,0,0,0,1,0,1,1,0,1,1,0,0/

      DATA ILP5/1,3,5,7,33,35,37,39,2,4,6,8,34,36,38,40,
     1       9,11,13,15,41,43,45,47,10,12,14,16,42,44,46,48,
     2       17,19,21,23,49,51,53,55,18,20,22,24,50,52,54,56,
     3       25,27,29,31,57,59,61,63,26,28,30,32,58,60,62,64/


      KLK1=0
      DO 40 I=1,16
      DO 30 J=1,2
      ILKUP1(I,J)=ILP1(KLK1+J)
      ILKUP4(I,J)=ILP4(KLK1+J)
   30  CONTINUE
      KLK1=KLK1+2
   40  CONTINUE

      KLK2=0
      DO 60 I=1,8
      I0=I-1
      DO 50 J=1,4
      J0=J-1
      IARRY=KLK2+J
      ILKUP2(I0,J0)=ILP2(IARRY)
      ILKUP3(I0,J0)=ILP3(IARRY)
      PLIUP1(I,J)=ROOT2*COS(PI*ILP3(IARRY)/4.0)
      PLIUP2(I,J)=ROOT2*SIN(PI*ILP3(IARRY)/4.0)
   50  CONTINUE
      KLK2=KLK2+4
   60  CONTINUE

      KLK3=0
      DO 80 I=1,8
      DO 70 J=1,8
      ILKUP5(I,J)=ILP5(KLK3+J)
   70  CONTINUE
      KLK3=KLK3+8
   80  CONTINUE


C     OPEN RELEVANT FILES

      OPEN(18,FILE='FDIMP',FORM='UNFORMATTED',IOSTAT=IOPT1)
      OPEN(KLOUT,FILE='OUTPUT',IOSTAT=IOPT2)
```

```
      WRITE(KLOUT,95)IOPT1,IOPT2
95    FORMAT(' ERRORS IN EXECUTION OF OPEN STATEMENT=',2I5)

      CALL G05CBF(IQ)

C     PREPARE FOR TRANSMISSION BY FIXING AN SNR VALUE
      DO 9000 IVAL=1,NOSNR
      SNR=SNR+SNRD
      STDVN=10.0**(-SNR/20.0)

      XX=0.0
      ICOUNT=0

      IERSYM=0
      IER1=0
      IER2=0
      IDER1=0
      IDER2=0

      MPBIT1=0
      MPBIT2=0

C     INITIALISING SIGNAL, NOISE ARRAYS AND STORED VECTORS

      DO 100 I=1,N
      IBIT1(I)=0
      IBIT2(I)=0
      IDBIT1(I)=0
      IDBIT2(I)=0
      SR(I)=ROOT2
      SI(I)=0.0
      ISTE(I)=0
      IVOUT(I)=0
100   CONTINUE

      DO 120 I=1,M
      COST(I)=10000.0
      DO 115 J=1,N
      VSR(I,J)=ROOT2
      VSI(I,J)=0.0
      ITX1(I,J)=0
      ITX2(I,J)=0
115   CONTINUE
120   CONTINUE
      COST(1)=0.0

      DO 140 I=1,NIMP
      HFR(I)=0.0
      HFI(I)=0.0
140   CONTINUE

C     ENTERING MAIN LOOP

      DO 8000 IMAIN=1,KLOOP
```

```
      ICOUNT=ICOUNT+1
      COUNT=REAL(ICOUNT)


C     INPUT SAMPLED IMPULSE RESPONSE

      READ(18)(YTR(I),I=1,NIMP)
      READ(18)(YTI(I),I=1,NIMP)

      YDN=YTR(1)*YTR(1)+YTI(1)*YTI(1)
      DO 146 I=1,NIMP
      YR(I)=(YTR(I)*YTR(1)+YTI(I)*YTI(1))/YDN
      YI(I)=(YTI(I)*YTR(1)-YTR(I)*YTI(1))/YDN
  146 CONTINUE


C     SHIFTING REGISTERS FOR INCOMING DATA SIGNAL

      DO 150 I=1,N1
      IX1=I+1
      IBIT1(I)=IBIT1(IX1)
      IBIT2(I)=IBIT2(IX1)
      IDBIT1(I)=IDBIT1(IX1)
      IDBIT2(I)=IDBIT2(IX1)
      SR(I)=SR(IX1)
      SI(I)=SI(IX1)
      ISTE(I)=ISTE(IX1)
      IVOUT(I)=IVOUT(IX1)
  150 CONTINUE

      DO 170 I=1,M
      DO 165 J=1,N1
      JX1=J+1
      VSR(I,J)=VSR(I,JX1)
      VSI(I,J)=VSI(I,JX1)
      ITX1(I,J)=ITX1(I,JX1)
      ITX2(I,J)=ITX2(I,JX1)
  165 CONTINUE
  170 CONTINUE


C     CALCULATING THE INTERSYMBOL INTERFERENCE

      DO 200 I=1,M
      GR(I)=0.0
      GI(I)=0.0
      DO 195 J=2,NIMP
      JZ=N+1-J
      GR(I)=GR(I)+VSR(I,JZ)*YR(J)-VSI(I,JZ)*YI(J)
      GI(I)=GI(I)+VSR(I,JZ)*YI(J)+VSI(I,JZ)*YR(J)
  195 CONTINUE
  200 CONTINUE


C     GENERATING SIGNAL IN BIT FORM

      XX=G05CAF(XX)
      IF(XX-0.5)210,210,220
  210 IBIT1(N)=0
      GO TO 230
  220 IBIT1(N)=1
  230 XX=G05CAF(XX)
      IF(XX-0.5)240,240,250
  240 IBIT2(N)=0
```

```
      GO TO 260
250   IBIT2(N)=1
260   CONTINUE

C     OBTAINING DIFFERENTIALLY PRE-CODED SIGNAL

      JEN1=IDBIT1(N1)*8+IDBIT2(N1)*4+IBIT1(N)*2+IBIT2(N)+1

      IDBIT1(N)=ILKUP1(JEN1,1)
      IDBIT2(N)=ILKUP1(JEN1,2)

C     OBTAINING CONVOLUTIONALLY CODED SIGNAL

      LEN1=ISTE(N1)
      LEN2=IDBIT1(N)*2+IDBIT2(N)

      ISTE(N)=ILKUP2(LEN1,LEN2)
      IVOUT(N)=ILKUP3(LEN1,LEN2)

      SR(N)=ROOT2*COS(PI*IVOUT(N)/4.0)
      SI(N)=ROOT2*SIN(PI*IVOUT(N)/4.0)

C     GENERATION OF NOISE

      DO 350 LNM=1,2
        DO 330 I=1,NIMP-1
        JX4=I+1
        HFR(I)=HFR(JX4)
        HFI(I)=HFI(JX4)
330     CONTINUE
      HFR(NIMP)=G05DDF(0.0,STDVN)
      HFI(NIMP)=G05DDF(0.0,STDVN)
      WR=0.0
      WI=0.0
        DO 340 I=1,NIMP
        JX5=NIMP-I+1
        WR=WR+HFR(JX5)*WFR(I)-HFI(JX5)*WFI(I)
        WI=WI+HFR(JX5)*WFI(I)+HFI(JX5)*WFR(I)
340     CONTINUE
350   CONTINUE
      WWR=(WR*YTR(1)+WI*YTI(1))/YDN
      WWI=(WI*YTR(1)-WR*YTI(1))/YDN

C     CALCULATION OF RECEIVED SIGNAL

      RR=0.0
      RI=0.0
      DO 400 I=1,NIMP
      JY=N+1-I
      RR=RR+SR(JY)*YR(I)-SI(JY)*YI(I)
      RI=RI+SR(JY)*YI(I)+SI(JY)*YR(I)
400   CONTINUE
      RR=RR+WWR
      RI=RI+WWI

C     FORMING 4M EXPANDED VECTORS AND AND CALCULATING THEIR COSTS

      KY=1
      JD1=0
```

```
      DO 500 I=1,M
      DO 480 JKL=1,4
      JD1=JD1+1
      DO 450 J=1,N1
      TR(JD1,J)=VSR(I,J)
      TI(JD1,J)=VSI(I,J)
      JTX1(JD1,J)=ITX1(I,J)
      JTX2(JD1,J)=ITX2(I,J)
450   CONTINUE

      TR(JD1,N)=PLIUP1(KY,JKL)
      DR=RR-GR(I)-TR(JD1,N)
      TI(JD1,N)=PLIUP2(KY,JKL)
      DI=RI-GI(I)-TI(JD1,N)

      JTX1(JD1,N)=ILKUP1(JKL,1)
      JTX2(JD1,N)=ILKUP1(JKL,2)

      CT1(JD1)=COST(I)+DR*DR+DI*DI

480   CONTINUE

      IF(MOD(I,MDM).EQ.0) KY=KY+1

500   CONTINUE


C     CHOOSING BEST VECTOR

      CC=1000000.0
      DO 510 I=1,M4
      IF(CT1(I)-CC)505,510,510
505   CC=CT1(I)
      MNO=I
510   CONTINUE


C     DISCARDING VECTORS WHICH DISAGREE

      DO 515 I=1,M4
      LMTR(I)=INT(TR(I,1)*10.0)
      LMTI(I)=INT(TI(I,1)*10.0)
515   CONTINUE
      IDETR=LMTR(MNO)
      IDETI=LMTI(MNO)

      DO 530 I=1,M4
      IF(LMTR(I)-IDETR)525,520,525
520   IF(LMTI(I)-IDETI)525,530,525
525   CT1(I)=100000.0
530   CONTINUE


C     SUBTRACTING THE SMALLEST COST FROM ALL THE COSTS

      CXY=CT1(MNO)
      DO 535 I=1,M4
      CT1(I)=CT1(I)-CXY
535   CONTINUE


C     OBTAINING BIT VALUE OF DETECTED SIGNAL

      MDBIT1=JTX1(MNO,1)
```

```
      MDBIT2=JTX2(MNO,1)

C     DIFFERENTIAL DECODING

      JEN2=MPBIT1*8+MPBIT2*4+MDBIT1*2+MDBIT2+1

      MBIT1=ILKUP4(JEN2,1)
      MBIT2=ILKUP4(JEN2,2)

C     ERROR COUNT

      IF(ICOUNT.LT.NTR)GO TO 545
      IF(INT(SR(1)*10.0)-IDETR)542,541,542
541   IF(INT(SI(1)*10.0)-IDETI)542,543,542
542   IERSYM=IERSYM+1
543   IF(IDBIT1(1).NE.MDBIT1)IER1=IER1+1
      IF(IDBIT2(1).NE.MDBIT2)IER2=IER2+1
      IF(IBIT1(1).NE.MBIT1)IDER1=IDER1+1
      IF(IBIT2(1).NE.MBIT2)IDER2=IDER2+1
545   CONTINUE

C     CHOOSING VECTORS FOR NEXT DETECTION PROCESS

      DO 830 I=1,MM
      DO 820 IJ1=1,MDM
      CC=1000000.0
         DO 810 J=1,ML1
         MDOT=ILKUP5(I,J)
         IF(CT1(MDOT)-CC)805,810,810
805         CC=CT1(MDOT)
          LG1=MDOT
810         CONTINUE
      CT2(I,IJ1)=CT1(LG1)
      LST1(I,IJ1)=LG1
      CT1(LG1)=10000000.0
820   CONTINUE
830   CONTINUE

C     TRANSFERRING THE SELECTED VECTORS TO THEIR ORIGINAL STORES
C     WITHOUT ALTERING THE POSITIONS CORRESPONDING TO THE COSTS.

      KZ=1
      DO 850 I=1,MM
      DO 845 IJ2=1,MDM
      JCHS=LST1(I,IJ2)
         DO 840 J=1,N
         VSR(KZ,J)=TR(JCHS,J)
         VSI(KZ,J)=TI(JCHS,J)
         ITX1(KZ,J)=JTX1(JCHS,J)
         ITX2(KZ,J)=JTX2(JCHS,J)
840         CONTINUE
      COST(KZ)=CT2(I,IJ2)
      KZ=KZ+1
845   CONTINUE
850   CONTINUE


      MPBIT1=MDBIT1
      MPBIT2=MDBIT2
```

```
8000  CONTINUE

C     CONFIRMATION OF SNR VALUE AND CALCULATION OF ERROR RATE

      SNRCALC=10.0*LOG10(1.0/(STDVN*STDVN))

      ERSYM=IERSYM/(COUNT-NTR)
      ERBIT=(IER1+IER2)/(2.0*(COUNT-NTR))
      ERDBIT=(IDER1+IDER2)/(2.0*(COUNT-NTR))

C     PRINTING IMPORTANT RESULTS
      WRITE(KLOUT,8100)KLOOP,NTR,IQ
8100  FORMAT(' KLOOP=',I10/' NTR=',I10/' IQ=',I10)
      WRITE(KLOUT,8110)ICOUNT,COUNT
8110  FORMAT(' ICOUNT=',I10/' COUNT=',F12.2)
      WRITE(KLOUT,8120)MM,N,NIMP,M
8120  FORMAT(' NO.OF STATES=',I10/' DELAY IN DETECTION=',I10/
     1       ' NO. OF COMPONENTS IN SIR =',I10/
     2       ' NO. OF STORED VECTORS    =',I10/)

      WRITE(KLOUT,8130)SNRCALC
8130  FORMAT(' SIGNAL-TO-NOISE RATIO=',F9.3,' DB')
      WRITE(KLOUT,8135)ERSYM
8135  FORMAT(' SYMBOL ERROR RATE                    =',E20.10)
      WRITE(KLOUT,8140)ERBIT
8140  FORMAT(' BIT ERROR RATE WITHOUT DIFFERENTIAL CODING =',E20.10)
      WRITE(KLOUT,8150)ERDBIT
8150  FORMAT(' BIT ERROR RATE WITH DIFFERENTIAL CODING    =',E20.10//)

      REWIND(UNIT=18)

9000  CONTINUE

      STOP
      END
####S
```

APPENDIX I

SIMULATION OF UNBIASED ESTIMATOR WITH DEGREE-1 PREDICTION AND PERFECT DETECTION

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION CF(5),Q(4,3000)
      DOUBLE PRECISION G05DDF

      DOUBLE PRECISION RR(-1:1),RI(-1:1)
      INTEGER ISNVR(-1:1),ISNVI(-1:1),ISSNVR(-1:1),ISSNVI(-1:1)
      DOUBLE PRECISION ASR(-1:1),ASI(-1:1),ASSR(-1:1),ASSI(-1:1)
      INTEGER IASR(-1:1),IASI(-1:1),IASSR(-1:1),IASSI(-1:1)

C     VALUES TO VARIABLES AND ARRAYS

      DATA CF /-0.1130119879D+01,0.3450267457D+00,-0.1166576134D+01,
     1        0.4780676433D+00,-0.5605390022D+00/
      DCG=0.05734208236D+00
      NOSAM=2400
      INFD=9


C     INITIALISING

      DL1=0.0
      DL2=0.0
      DL3=0.0
      DL4=0.0
      DL5=0.0

      JQ=50+NOSAM
      JQ1=JQ+1


      CALL G05CBF(INFD)

C     GENERATION OF Q1(T) AND Q2(T)

      DO 250 I=1,4

      JA=1

      TF3=0.0
      TVF3=0.0
      TF3DCG=0.0
      TVF3DCG=0.0


        DO 240 J=1,JQ1

        F0=G05DDF(0.0D0,1.0D0)
        F1=F0-(DL1*CF(1)+DL2*CF(2))
        F2=F1-(DL3*CF(3)+DL4*CF(4))
        F3=F2-(DL5*CF(5))
        F3DCG=F3*DCG

        DL5=F3
        DL4=DL3
```

```
      DL3=F2
      DL2=DL1
      DL1=F1

      IF(J.LE.50) GO TO 240

      Q(I,JA)=F3DCG
      JA=JA+1


      TF3=TF3+F3
      TVF3=TVF3+(F3*F3)
      TF3DCG=TF3DCG+F3DCG
      TVF3DCG=TVF3DCG+(F3DCG*F3DCG)


240       CONTINUE


      EF3=TF3/(NOSAM+1)
      VARF3=TVF3/(NOSAM+1)
      EF3DCG=TF3DCG/(NOSAM+1)
      VARF3DCG=TVF3DCG/(NOSAM+1)


      WRITE(0,244)EF3,VARF3
      WRITE(0,246)EF3DCG,VARF3DCG
244   FORMAT("MEAN OF F3 =",E17.10,5X,"VARIANCE OF F3 =",E17.10)
246   FORMAT("MEAN OF F3DCG =",E17.10,2X,"VARIANCE OF F3DCG =",
     1 E17.10)

250   CONTINUE

C     INITIALISING VARIABLES FOR MAIN PROGRAM

      IFRB=960
      IFR=9600
      IFDIV=IFR/IFRB
      FDIV=REAL(IFDIV)
      NOD=24000
      NTR=2400
      INT=NOD/IFDIV

      INDTN=10

      SNR=30.0
      SIGSQN=0.5*(10.0**(-SNR/10.0))
      STDVN=SIGSQN**0.5

      NOFL=20
      NOTH=1
      NOB1=1
      THETA=0.0
      B1FST=1.0
      B1=B1FST
      THD=0.05
      B1D=-0.1
```

```
      DO 6000 LT1=1,NOTH

      WRITE(NOFL,475)THETA
475   FORMAT(7H"THETA=,F9.4,1H")


      DO 5000 LT2=1,NOB1

      TQR=0.0
      TVQR=0.0
      TQI=0.0
      TVQI=0.0
      TQQR=0.0
      TVQQR=0.0
      TQQI=0.0
      TVQQI=0.0

      XX=0.0
      ERTOT=0.0
      ICOUNT=0

      YP2R=0.0
      YP2I=0.0
      XGRR=0.0
      XGRI=0.0

      CALL G05CBF(INDTN)


C     ENTERING MAIN LOOP

      DO 4000 IM=1,INT

      QA1=Q(1,IM)
      QB1=Q(1,IM+1)
      QA2=Q(2,IM)
      QB2=Q(2,IM+1)
      QA3=Q(3,IM)
      QB3=Q(3,IM+1)
      QA4=Q(4,IM)
      QB4=Q(4,IM+1)


      CON1=(QB1-QA1)/FDIV
      CON2=(QB2-QA2)/FDIV
      CON3=(QB3-QA3)/FDIV
      CON4=(QB4-QA4)/FDIV


C           ENTERING SECONDARY LOOP

      DO 3000 ISEC=1,IFDIV

      ICOUNT=ICOUNT+1

      QR=QA1+(ISEC-1)*CON1
      QI=QA2+(ISEC-1)*CON2
      QQR=QA3+(ISEC-1)*CON3
      QQI=QA4+(ISEC-1)*CON4

      TQR=TQR+QR
      TVQR=TVQR+(QR*QR)
      TQI=TQI+QI
      TVQI=TVQI+(QI*QI)
```

```
      TQQR=TQQR+QQR
      TVQQR=TVQQR+(QQR*QQR)
      TQQI=TQQI+QQI
      TVQQI=TVQQI+(QQI*QQI)

      YR=QR
      YI=-QI
      YYR=QQR
      YYI=-QQI


C     GENERATING SIGNAL

      XX=G05CAF(XX)
      IF(XX-0.25)740,740,710
710     IF(XX-0.75)770,770,760
740     SR=-1.0
      ISR=-1
      SI=0.0
      ISI=0
      GO TO 800
760     SR=1.0
      ISR=1
      SI=0.0
      ISI=0
      GO TO 800

770     SR=0.0
      ISR=0
      XX=G05CAF(XX)
      IF(XX-0.5)780,780,790
780     SI=-1.0
      ISI=-1
      GO TO 800
790     SI=1.0
      ISI=1
      GO TO 800

800     XX=G05CAF(XX)
      IF(XX-0.25)840,840,810
810     IF(XX-0.75)870,870,860
840     SSR=-1.0
      ISSR=-1
      SSI=0.0
      ISSI=0
      GO TO 900
860     SSR=1.0
      ISSR=1
      SSI=0.0
      ISSI=0
      GO TO 900

870     SSR=0.0
      ISSR=0
      XX=G05CAF(XX)
      IF(XX-0.5)880,880,890
880     SSI=-1.0
      ISSI=-1
      GO TO 900
890     SSI=1.0
      ISSI=1
      GO TO 900

900     CONTINUE
```

```
C     GENERATING WHITE GAUSSIAN NOISE

      WR=G05DDF(0.0D00,STDVN)
      WI=G05DDF(0.0D00,STDVN)


C     SAVING THE FIRST TWO RECEIVED SAMPLES AND RELEVANT QUANTITIES

      IF(ICOUNT-2)1100,1200,1300

1100  RR(-1)=(SR*YR-SI*YI)+(SSR*YYR-SSI*YYI)+WR
      RI(-1)=(SI*YR+SR*YI)+(SSI*YYR+SSR*YYI)+WI
      ASR(-1)=SR
      ASI(-1)=SI
      ASSR(-1)=SSR
      ASSI(-1)=SSI
      IASR(-1)=ISR
      IASI(-1)=ISI
      IASSR(-1)=ISSR
      IASSI(-1)=ISSI
      DNM=ISR*ISR+ISI*ISI
      DDNM=ISSR*ISSR+ISSI*ISSI
      ISNVR(-1)=ISR/DNM
      ISNVI(-1)=-ISI/DNM
      ISSNVR(-1)=ISSR/DDNM
      ISSNVI(-1)=-ISSI/DDNM
      GO TO 3000

1200  RR(0)=(SR*YR-SI*YI)+(SSR*YYR-SSI*YYI)+WR
      RI(0)=(SI*YR+SR*YI)+(SSI*YYR+SSR*YYI)+WI
      ASR(0)=SR
      ASI(0)=SI
      ASSR(0)=SSR
      ASSI(0)=SSI
      IASR(0)=ISR
      IASI(0)=ISI
      IASSR(0)=ISSR
      IASSI(0)=ISSI
      DNM=ISR*ISR+ISI*ISI
      DDNM=ISSR*ISSR+ISSI*ISSI
      ISNVR(0)=ISR/DNM
      ISNVI(0)=-ISI/DNM
      ISSNVR(0)=ISSR/DDNM
      ISSNVI(0)=-ISSI/DDNM

      YP1R=YR
      YP1I=YI
      GO TO 3000


C     CALCULATING ERRORS

1300  IF(ICOUNT.LT.NTR)GO TO 1350
      EXR=YR-YP2R
      EXI=YI-YP2I
      ERXY=EXR*EXR+EXI*EXI
      ERTOT=ERTOT+ERXY


C           CALCULATING RECEIVED SIGNAL
```

```
1350      RR(1)=(SR*YR-SI*YI)+(SSR*YYR-SSI*YYI)+WR
          RI(1)=(SI*YR+SR*YI)+(SSI*YYR+SSR*YYI)+WI
      ASR(1)=SR
      ASI(1)=SI
      ASSR(1)=SSR
      ASSI(1)=SSI
      IASR(1)=ISR
      IASI(1)=ISI
      IASSR(1)=ISSR
      IASSI(1)=ISSI


C     CALCULATING SIGNAL INVERSE

      DNM=ISR*ISR+ISI*ISI
      DDNM=ISSR*ISSR+ISSI*ISSI
      ISNVR(1)=ISR/DNM
      ISNVI(1)=-ISI/DNM
      ISSNVR(1)=ISSR/DDNM
      ISSNVI(1)=-ISSI/DDNM


C     CALCULATION OF INTERMEDIATE VALUES LEADING TO ESTIMATE OF CHANNEL

      REQR=(2*(ISSNVR(0)*RR(0)-ISSNVI(0)*RI(0)))-
     1     (ISSNVR(-1)*RR(-1)-ISSNVI(-1)*RI(-1))-
     1     (ISSNVR(1)*RR(1)-ISSNVI(1)*RI(1))
      REQI=(2*(ISSNVI(0)*RR(0)+ISSNVR(0)*RI(0)))-
     1     (ISSNVI(-1)*RR(-1)+ISSNVR(-1)*RI(-1))-
     1     (ISSNVI(1)*RR(1)+ISSNVR(1)*RI(1))

      IEQLR=(ISSNVR(-1)*IASR(-1)-ISSNVI(-1)*IASI(-1))-
     1     (ISSNVR(1)*IASR(1)-ISSNVI(1)*IASI(1))
      IEQLI=(ISSNVI(-1)*IASR(-1)+ISSNVR(-1)*IASI(-1))-
     1     (ISSNVI(1)*IASR(1)+ISSNVR(1)*IASI(1))

      EQMR=(IEQLR*XGRR)-(IEQLI*XGRI)
      EQMI=(IEQLI*XGRR)+(IEQLR*XGRI)

      ENUMR=REQR-EQMR
      ENUMI=REQI-EQMI

      IAEQR=(2*(ISSNVR(0)*IASR(0)-ISSNVI(0)*IASI(0)))-
     1     (ISSNVR(-1)*IASR(-1)-ISSNVI(-1)*IASI(-1))-
     1     (ISSNVR(1)*IASR(1)-ISSNVI(1)*IASI(1))
      IAEQI=(2*(ISSNVI(0)*IASR(0)+ISSNVR(0)*IASI(0)))-
     1     (ISSNVI(-1)*IASR(-1)+ISSNVR(-1)*IASI(-1))-
     1     (ISSNVI(1)*IASR(1)+ISSNVR(1)*IASI(1))

      IADBSQ=IAEQR*IAEQR+IAEQI*IAEQI
      ADBSQ=REAL(IADBSQ)
      ADBSQM=ADBSQ**0.5


C     PREDICTION PROCESS BEGINS

      IF(IADBSQ.NE.0)GO TO 1400
      EPSR=0.0
      EPSI=0.0
      GO TO 1500

1400  Y1ESR=(ENUMR*IAEQR+ENUMI*IAEQI)/ADBSQ
      Y1ESI=(ENUMI*IAEQR-ENUMR*IAEQI)/ADBSQ
      ER=Y1ESR-YP1R
```

```
      EI=Y1ESI-YP1I
      EPSR=ADBSQM*ER*B1
      EPSI=ADBSQM*EI*B1

1500  XGRR=XGRR+((1.0-THETA)**2)*EPSR
      XGRI=XGRI+((1.0-THETA)**2)*EPSI

      YP1R=YP1R+XGRR+(1.0-(THETA*THETA))*EPSR
      YP1I=YP1I+XGRI+(1.0-(THETA*THETA))*EPSI

      YP2R=YP1R+XGRR
      YP2I=YP1I+XGRI

C     SHIFTING OF ARRAYS

      IV=-2
      DO 1600 I=1,2
           IV=IV+1
      RR(IV)=RR(IV+1)
      RI(IV)=RI(IV+1)
      ASR(IV)=ASR(IV+1)
      ASI(IV)=ASI(IV+1)
      ASSR(IV)=ASSR(IV+1)
      ASSI(IV)=ASSI(IV+1)
      IASR(IV)=IASR(IV+1)
      IASI(IV)=IASI(IV+1)
      IASSR(IV)=IASSR(IV+1)
      IASSI(IV)=IASSI(IV+1)
      ISNVR(IV)=ISNVR(IV+1)
      ISNVI(IV)=ISNVI(IV+1)
      ISSNVR(IV)=ISSNVR(IV+1)
      ISSNVI(IV)=ISSNVI(IV+1)
1600  CONTINUE


3000       CONTINUE


4000  CONTINUE

      SNRCALC=10*LOG10(0.5/SIGSQN)

      EXQ1=TQR/ICOUNT
      EXQ2=TQI/ICOUNT
      VARQ1=TVQR/ICOUNT
      VARQ2=TVQI/ICOUNT
      EXQQ1=TQQR/ICOUNT
      EXQQ2=TQQI/ICOUNT
      VARQQ1=TVQQR/ICOUNT
      VARQQ2=TVQQI/ICOUNT

      ERRORP2=10*LOG10(ERTOT/(ICOUNT-NTR))

      WRITE(NOFL,4005)B1,ERRORP2
4005  FORMAT(F8.4,",",E20.10)


C     PRINTING OUT IMPORTANT RESULTS

      WRITE(0,4007)NOSAM,SNRCALC
4007  FORMAT("NOSAM=",I7,2X,"SNR=",F8.5)
      WRITE(0,4010)THETA,B1
```

```
4010  FORMAT("THETA=",F10.4/"B1=",F10.4)

      WRITE(0,4020)NOD,IFRB,IFR,ICOUNT
4020  FORMAT("NOD=",I7,2X,"IFRB=",I7,2X,"IFR=",I7,2X,"ICOUNT=",I7)
      WRITE(0,4030)INFD,INDTN
4030  FORMAT("INFD=",I3,2X,"INDTN=",I3)

      WRITE(0,4040)EXQ1,VARQ1
4040  FORMAT("MEAN OF Q1 =",E17.10,2X,"VARIANCE OF Q1 =",E17.10)
      WRITE(0,4050)EXQ2,VARQ2
4050  FORMAT("MEAN OF Q2 =",E17.10,2X,"VARIANCE OF Q2 =",E17.10)

      WRITE(0,4060)EXQQ1,VARQQ1
4060  FORMAT("MEAN OF QQ1 =",E17.10,2X,"VARIANCE OF QQ1 =",E17.10)
      WRITE(0,4070)EXQQ2,VARQQ2
4070  FORMAT("MEAN OF QQ2 =",E17.10,2X,"VARIANCE OF QQ2 =",E17.10)

      WRITE(0,4080)ERRORP2
4080  FORMAT("2ND STEP PREDICTION ERROR IN CHANNEL 1=",E20.10)


      B1=B1+B1D

5000  CONTINUE

      THETA=THETA+THD
      B1=B1FST

6000  CONTINUE


      STOP
      END
```

APPENDIX J

SIMULATION OF COMBINED DETECTOR AND MODIFIED GRADIENT ESTIMATOR FOR FAST FADING CHANNELS

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION CF(5),Q(4,3000)
      DOUBLE PRECISION G05DDF
      DOUBLE PRECISION A(16),B(16),C(16),D(16)
      DOUBLE PRECISION SR(32),SI(32),SSR(32),SSI(32)
      DOUBLE PRECISION XR(4,32),XI(4,32),XXR(4,32),XXI(4,32)
      DOUBLE PRECISION YP1R(4,32),YP1I(4,32),YYP1R(4,32),YYP1I(4,32)
      DOUBLE PRECISION XGRR(4,32),XGRI(4,32),XGR2R(4,32),XGR2I(4,32)
      DOUBLE PRECISION GR(10,31),GI(10,31),GGR(10,31),GGI(10,31)
      DOUBLE PRECISION GXR(10,31),GXI(10,31),GGXR(10,31),GGXI(10,31)
      DOUBLE PRECISION COST(4),CT(10),CTR(16)
      INTEGER IA(16),IB(16),IC(16),ID(16)
      INTEGER ISR(32),ISI(32),ISSR(32),ISSI(32)
      INTEGER IXR(4,32),IXI(4,32),IXXR(4,32),IXXI(4,32)
      INTEGER ITR(10,32),ITI(10,32),ITTR(10,32),ITTI(10,32)
      INTEGER INN(10),INZ(4)
      INTEGER IHH(32),IDIFCD(16,2)
      INTEGER IENCD(4,32),IENDIF(4,32),IDECD(4,0:1),IDCDIF(4)


C     INITIALISING VARIABLES FOR FADING GENERATOR

      DATA CF /-0.1130119879D+01,0.3450267457D+00,-0.1166576134D+01,
     1        0.4780676433D+00,-0.5605390022D+00/
      DCG=0.05734208236D+00
      NOSAM=2400
      INFD=9

      DL1=0.0
      DL2=0.0
      DL3=0.0
      DL4=0.0
      DL5=0.0

      JQ=50+NOSAM
      JQ1=JQ+1

      CALL G05CBF(INFD)


C     FADING GENERATOR

      DO 250 I=1,4

      JA=1

      TF3DCG=0.0
      TVF3DCG=0.0


        DO 240 J=1,JQ1

        F0=G05DDF(0.0D0,1.0D0)
        F1=F0-(DL1*CF(1)+DL2*CF(2))
        F2=F1-(DL3*CF(3)+DL4*CF(4))
        F3=F2-(DL5*CF(5))
        F3DCG=F3*DCG
```

```
DL5=F3
DL4=DL3
DL3=F2
DL2=DL1
DL1=F1

IF(J.LE.50) GO TO 240

Q(I,JA)=F3DCG
JA=JA+1

TF3DCG=TF3DCG+F3DCG
TVF3DCG=TVF3DCG+(F3DCG**2)

240      CONTINUE

    EF3DCG=TF3DCG/(NOSAM+1)
    VARF3DCG=TVF3DCG/(NOSAM+1)

    WRITE(0,244)I,EF3DCG
    WRITE(0,246)I,VARF3DCG
244   FORMAT("MEAN OF Q",I1,"(T)   =",E16.7)
246   FORMAT("VARIANCE OF Q",I1,"(T)=",E16.7)

250   CONTINUE

    PRINT *,"      "



C     INITIALISING VARIABLES FOR MAIN PROGRAM

    IRETR=50
    IFDIV=10
    FDIV=REAL(IFDIV)
    NOD=24000
    NTR=NOD/10
    RNOD=REAL(NOD)
    INTT=NOD/IFDIV
    NOSNR=4

    SNR=10.0
    SNRD=2.5
    SNR=SNR-SNRD

    INDTN=80

    BCONST=0.55
    THETA=0.79
    PSI=1.0

    K=4
    N=32
    N1=N-1

    DATA (IA(I),I=1,8)/8*1/,(IA(I),I=9,16)/8*-1/
    DATA (IB(I),IB(I+1),IB(I+2),IB(I+3),I=1,9,8)/8*1/,
   1     (IB(I),IB(I+1),IB(I+2),IB(I+3),I=5,13,8)/8*-1/
    DATA (IC(I),IC(I+1),I=1,13,4)/8*1/,
   1     (IC(I),IC(I+1),I=3,15,4)/8*-1/
    DATA (ID(I),I=1,15,2)/8*1/,(ID(I),I=2,16,2)/8*-1/

    DATA IHH/0,0,0,1,1,0,1,1,1,0,0,0,1,1,0,1,
```

```
     1       0,1,1,1,0,0,1,0,1,1,1,0,0,1,0,0/

        DO 400 I=1,16
        A(I)=REAL(IA(I))
        B(I)=REAL(IB(I))
        C(I)=REAL(IC(I))
        D(I)=REAL(ID(I))
400     CONTINUE

        KFC=0
        DO 600 I=1,16
        DO 550 J=1,2
        IDIFCD(I,J)=IHH(KFC+J)
550     CONTINUE
        KFC=KFC+2
600     CONTINUE


C       PREPARE FOR TRANSMISSION BY FIXING AN SNR VALUE

        DO 6000 IVAL=1,NOSNR

        CALL G05CBF(INDTN)

        SNR=SNR+SNRD
        P=10.0**(-SNR/20.0)
        SIGSQN=P**2

        ICOUNT=0
        COUNT=0.0
        KCOUNT=0
        DO 650 I=1,4
        IDECD(I,0)=0
650     CONTINUE

        EQR=0.0
        VQR=0.0
        EQI=0.0
        VQI=0.0
        EQQR=0.0
        VQQR=0.0
        EQQI=0.0
        VQQI=0.0

        XX=0.0

        IER1SYM=0
        IER2SYM=0
        IER1=0
        IER2=0
        IER3=0
        IER4=0
        IDER1=0
        IDER2=0
        IDER3=0
        IDER4=0


C       INITIALISING SIGNAL ARRAYS

        DO 660 I=1,N
        SR(I)=1.0
        SI(I)=1.0
```

```
      SSR(I)=1.0
      SSI(I)=1.0
      ISR(I)=1
      ISI(I)=1
      ISSR(I)=1
      ISSI(I)=1
660   CONTINUE


C     INITIALISING STORED VECTORS AND COSTS
C     ALSO INITIALISING ARRAYS CONTAINING BIT VALUES OF SIGNAL

      DO 670 I=1,K
      COST(I)=1000.0
      DO 665 J=1,N
      XR(I,J)=1.0
      XI(I,J)=1.0
      XXR(I,J)=1.0
      XXI(I,J)=1.0
      IXR(I,J)=1
      IXI(I,J)=1
      IXXR(I,J)=1
      IXXI(I,J)=1

      YP1R(I,J)=0.9814489005D+00
      YP1I(I,J)=-0.5665125350D+00
      YYP1R(I,J)=-0.6796669331D+00
      YYP1I(I,J)=0.2566136036D+00
      XGRR(I,J)=0.0005686099D+00
      XGRI(I,J)=0.0241126376D+00
      XGR2R(I,J)=-0.0134283878D+00
      XGR2I(I,J)=0.0153125377D+00

          IENCD(I,J)=0
          IENDIF(I,J)=0

665       CONTINUE
670   CONTINUE

      COST(1)=0.0


C                ENTERING MAIN LOOP

                 DO 4000 IM=1,INTT

                 CON1=(Q(1,IM+1)-Q(1,IM))/FDIV
                 CON2=(Q(2,IM+1)-Q(2,IM))/FDIV
                 CON3=(Q(3,IM+1)-Q(3,IM))/FDIV
                 CON4=(Q(4,IM+1)-Q(4,IM))/FDIV

C          ENTERING SECONDARY LOOP

                 DO 3000 ISEC=1,IFDIV

                 ICOUNT=ICOUNT+1
                 COUNT=REAL(ICOUNT)
                 KCOUNT=KCOUNT+1

                 QR=Q(1,IM)+((ISEC-1)*CON1)
                 QI=Q(2,IM)+((ISEC-1)*CON2)
                 QQR=Q(3,IM)+((ISEC-1)*CON3)
                 QQI=Q(4,IM)+((ISEC-1)*CON4)
```

```
EQR=EQR+QR
VQR=VQR+(QR**2)
EQI=EQI+QI
VQI=VQI+(QI**2)
EQQR=EQQR+QQR
VQQR=VQQR+(QQR**2)
EQQI=EQQI+QQI
VQQI=VQQI+(QQI**2)

YR=QR
YI=-QI
YYR=QQR
YYI=-QQI
      IF(KCOUNT.EQ.IRETR) THEN
      DO 678 I=1,K
      YP1R(I,N)=YR
      YP1I(I,N)=YI
      YYP1R(I,N)=YYR
      YYP1I(I,N)=YYI
      XGRR(I,N)=YR-PYR
      XGRI(I,N)=YI-PYI
      XGR2R(I,N)=YYR-PYYR
      XGR2I(I,N)=YYI-PYYI
678     CONTINUE
      KCOUNT=1
      END IF


C     SHIFTING SIGNAL REGISTERS FOR INCOMING DATA SYMBOL

      DO 680 I=1,N1
      JX=I+1
      SR(I)=SR(JX)
      SI(I)=SI(JX)
      SSR(I)=SSR(JX)
      SSI(I)=SSI(JX)
      ISR(I)=ISR(JX)
      ISI(I)=ISI(JX)
      ISSR(I)=ISSR(JX)
      ISSI(I)=ISSI(JX)
680   CONTINUE


C     SHIFTING STORED VECTORS TO ACCOMMODATE NEW VALUE
C     ALSO SHIFTING ARRAYS CONTAINING BIT VALUES OF SIGNAL

      DO 690 I=1,K
         DO 685 J=1,N1
         JY=J+1
         XR(I,J)=XR(I,JY)
         XI(I,J)=XI(I,JY)
         XXR(I,J)=XXR(I,JY)
         XXI(I,J)=XXI(I,JY)
         IXR(I,J)=IXR(I,JY)
         IXI(I,J)=IXI(I,JY)
         IXXR(I,J)=IXXR(I,JY)
         IXXI(I,J)=IXXI(I,JY)

         YP1R(I,J)=YP1R(I,JY)
         YP1I(I,J)=YP1I(I,JY)
         YYP1R(I,J)=YYP1R(I,JY)
         YYP1I(I,J)=YYP1I(I,JY)
```

```
      XGRR(I,J)=XGRR(I,JY)
      XGRI(I,J)=XGRI(I,JY)
      XGR2R(I,J)=XGR2R(I,JY)
      XGR2I(I,J)=XGR2I(I,JY)

      IENCD(I,J)=IENCD(I,JY)
      IENDIF(I,J)=IENDIF(I,JY)

685     CONTINUE
690   CONTINUE


C       GENERATING SIGNAL

700     XX=G05CAF(XX)
      IF(XX-0.5)710,710,720
710     SR(N)=-1.0
      ISR(N)=-1
      IENCD(2,N)=1
      GO TO 750
720     SR(N)=1.0
      ISR(N)=1
      IENCD(2,N)=0
      GO TO 750

750     XX=G05CAF(XX)
      IF(XX-0.5)760,760,770
760     SI(N)=-1.0
      ISI(N)=-1
      IENCD(1,N)=1
      GO TO 800
770     SI(N)=1.0
      ISI(N)=1
      IENCD(1,N)=0
      GO TO 800

800     XX=G05CAF(XX)
      IF(XX-0.5)810,810,820
810     SSR(N)=-1.0
      ISSR(N)=-1
      IENCD(4,N)=1
      GO TO 850
820     SSR(N)=1.0
      ISSR(N)=1
      IENCD(4,N)=0
      GO TO 850

850     XX=G05CAF(XX)
      IF(XX-0.5)860,860,870
860     SSI(N)=-1.0
      ISSI(N)=-1
      IENCD(3,N)=1
      GO TO 900
870     SSI(N)=1.0
      ISSI(N)=1
      IENCD(3,N)=0
      GO TO 900

900     CONTINUE


C   DIFFERENTIALLY ENCODING SIGNAL

    JENC1=IENCD(1,N1)*8+IENCD(2,N1)*4+IENCD(1,N)*2+IENCD(2,N)+1
```

```
      JENC2=IENCD(3,N1)*8+IENCD(4,N1)*4+IENCD(3,N)*2+IENCD(4,N)+1

      IENDIF(1,N)=IDIFCD(JENC1,1)
      IENDIF(2,N)=IDIFCD(JENC1,2)
      IENDIF(3,N)=IDIFCD(JENC2,1)
      IENDIF(4,N)=IDIFCD(JENC2,2)


C          CALCULATING RECEIVED SIGNAL

          WR=G05DDF(0.0D00,P)
          WI=G05DDF(0.0D00,P)
          RR=(SR(N)*YR-SI(N)*YI)+(SSR(N)*YYR-SSI(N)*YYI)+WR
          RI=(SI(N)*YR+SR(N)*YI)+(SSI(N)*YYR+SSR(N)*YYI)+WI


C     EXPANDING THE VECTORS ACCORDINGLY

          IBRAVO=0
          DO 1000 I=1,K
          DO 910 J=1,16
          DR=RR-(A(J)*YP1R(I,N1)-B(J)*YP1I(I,N1))
         1     -(C(J)*YYP1R(I,N1)-D(J)*YYP1I(I,N1))
          DI=RI-(B(J)*YP1R(I,N1)+A(J)*YP1I(I,N1))
         1     -(D(J)*YYP1R(I,N1)+C(J)*YYP1I(I,N1))
          CTR(J)=PSI*COST(I)+DR**2+DI**2
  910   CONTINUE
            DO 990 ITRAN=1,K-I+1
            CC=1000000.0
            DO 970 JTRAN=1,16
            IF(CTR(JTRAN)-CC)960,970,970
  960         CC=CTR(JTRAN)
            JNX=JTRAN
  970         CONTINUE
            CT(IBRAVO+ITRAN)=CC
            INN(IBRAVO+ITRAN)=JNX
            CTR(JNX)=10000000D00
  990         CONTINUE
        IBRAVO=IBRAVO+K-I+1
  1000  CONTINUE


C     TRANSFERRING THE 10 EXPANDED VECTORS TO A TEMPORARY STORE

        KB=0
        DO 1150 I=1,K
            DO 1140 KM=1,K-I+1
                DO 1130 J=1,N1
                ITR(KB+KM,J)=IXR(I,J)
                ITI(KB+KM,J)=IXI(I,J)
                ITTR(KB+KM,J)=IXXR(I,J)
                ITTI(KB+KM,J)=IXXI(I,J)
                GR(KB+KM,J)=YP1R(I,J)
                GI(KB+KM,J)=YP1I(I,J)
                GGR(KB+KM,J)=YYP1R(I,J)
                GGI(KB+KM,J)=YYP1I(I,J)
                GXR(KB+KM,J)=XGRR(I,J)
                GXI(KB+KM,J)=XGRI(I,J)
                GGXR(KB+KM,J)=XGR2R(I,J)
                GGXI(KB+KM,J)=XGR2I(I,J)
  1130            CONTINUE
            LAST=INN(KB+KM)
            ITR(KB+KM,N)=IA(LAST)
            ITI(KB+KM,N)=IB(LAST)
```

```
        ITTR(KB+KM,N)=IC(LAST)
        ITTI(KB+KM,N)=ID(LAST)
1140      CONTINUE
    KB=KB+K-I+1
1150  CONTINUE


C     SELECTING THE BEST VECTOR FROM THE 10 STORED VECTORS

    CC=1000000.0
    DO 1170 I=1,10
    IF(CT(I)-CC)1160,1170,1170
1160  CC=CT(I)
    ILUCK=I
1170  CONTINUE
    COST(1)=CC
    INZ(1)=ILUCK
    CT(ILUCK)=10000000D00


C     OBTAINING BIT VALUE OF DETECTED SIGNAL

    KADD=ITR(ILUCK,1)+ITI(ILUCK,1)
    KKADD=ITTR(ILUCK,1)+ITTI(ILUCK,1)

    IF(KADD-0)1171,1173,1172
1171  IDECD(1,1)=1
    IDECD(2,1)=1
    GO TO 1174
1172  IDECD(1,1)=0
    IDECD(2,1)=0
    GO TO 1174
1173  IF(ITR(ILUCK,1).EQ.1) THEN
    IDECD(1,1)=1
    IDECD(2,1)=0
    ELSE
    IDECD(1,1)=0
    IDECD(2,1)=1
    END IF
    GO TO 1174

1174  IF(KKADD-0)1175,1177,1176
1175  IDECD(3,1)=1
    IDECD(4,1)=1
    GO TO 1178
1176  IDECD(3,1)=0
    IDECD(4,1)=0
    GO TO 1178
1177  IF(ITTR(ILUCK,1).EQ.1) THEN
    IDECD(3,1)=1
    IDECD(4,1)=0
    ELSE
    IDECD(3,1)=0
    IDECD(4,1)=1
    END IF
    GO TO 1178


C     DIFFERENTIAL DECODING OPERATION

1178  JDEC1=IDECD(1,0)*8+IDECD(2,0)*4+IDECD(1,1)*2+IDECD(2,1)+1
    JDEC2=IDECD(3,0)*8+IDECD(4,0)*4+IDECD(3,1)*2+IDECD(4,1)+1

    IDCDIF(1)=IDIFCD(JDEC1,1)
```

```
      IDCDIF(2)=IDIFCD(JDEC1,2)
      IDCDIF(3)=IDIFCD(JDEC2,1)
      IDCDIF(4)=IDIFCD(JDEC2,2)


C     ERROR COUNT

      IF(ICOUNT.LT.NTR)GO TO 1190
      IF(ISR(1)-ITR(ILUCK,1)) 1182,1181,1182
1181  IF(ISI(1)-ITI(ILUCK,1)) 1182,1184,1182
1182  IER1SYM=IER1SYM+1
1184  CONTINUE
      IF(IENCD(1,1).NE.IDECD(1,1)) IER1=IER1+1
      IF(IENCD(2,1).NE.IDECD(2,1)) IER2=IER2+1
      IF(IENDIF(1,1).NE.IDCDIF(1)) IDER1=IDER1+1
      IF(IENDIF(2,1).NE.IDCDIF(2)) IDER2=IDER2+1

      IF(ISSR(1)-ITTR(ILUCK,1)) 1187,1186,1187
1186  IF(ISSI(1)-ITTI(ILUCK,1)) 1187,1189,1187
1187  IER2SYM=IER2SYM+1
1189  CONTINUE
      IF(IENCD(3,1).NE.IDECD(3,1)) IER3=IER3+1
      IF(IENCD(4,1).NE.IDECD(4,1)) IER4=IER4+1
      IF(IENDIF(3,1).NE.IDCDIF(3)) IDER3=IDER3+1
      IF(IENDIF(4,1).NE.IDCDIF(4)) IDER4=IDER4+1
1190  CONTINUE


C     SHIFT BIT VALUE OF DETECTED SIGNAL, FOR USE IN NEXT DETECTION

      DO 1195 I=1,4
      IDECD(I,0)=IDECD(I,1)
1195  CONTINUE


C     DISCARDING THE VECTORS THAT DO NOT AGREE

      DO 1230 I=1,10
      IF(ITR(ILUCK,1)-ITR(I,1))1225,1205,1225
1205  IF(ITI(ILUCK,1)-ITI(I,1))1225,1210,1225
1210  IF(ITTR(ILUCK,1)-ITTR(I,1))1225,1215,1225
1215  IF(ITTI(ILUCK,1)-ITTI(I,1))1225,1230,1225
1225  CT(I)=10000.0
1230  CONTINUE


C     SELECT (K-1) MORE VECTORS FROM THE REMAINDER

      DO 1250 I=1,K-1
      CC=1000000.0
          DO 1240 J=1,10
          IF(CT(J)-CC)1235,1240,1240
1235        CC=CT(J)
          JLUCK=J
1240        CONTINUE
      COST(I+1)=CC
      INZ(I+1)=JLUCK
      CT(JLUCK)=10000000D00
1250  CONTINUE


C     TRANSFER THE 4 SELECTED VECTORS BACK TO THE X STORE
```

```
      DO 1270 I=1,K
      IFIN=INZ(I)
          DO 1260 J=1,N
          IXR(I,J)=ITR(IFIN,J)
          IXI(I,J)=ITI(IFIN,J)
          IXXR(I,J)=ITTR(IFIN,J)
          IXXI(I,J)=ITTI(IFIN,J)
          XR(I,J)=REAL(IXR(I,J))
          XI(I,J)=REAL(IXI(I,J))
          XXR(I,J)=REAL(IXXR(I,J))
          XXI(I,J)=REAL(IXXI(I,J))
1260        CONTINUE
          DO 1265 J=1,N1
          YP1R(I,J)=GR(IFIN,J)
          YP1I(I,J)=GI(IFIN,J)
          YYP1R(I,J)=GGR(IFIN,J)
          YYP1I(I,J)=GGI(IFIN,J)
          XGRR(I,J)=GXR(IFIN,J)
          XGRI(I,J)=GXI(IFIN,J)
          XGR2R(I,J)=GGXR(IFIN,J)
          XGR2I(I,J)=GGXI(IFIN,J)
1265        CONTINUE
1270  CONTINUE


C     ESTIMATION PROCEDURE BEGINS

      DO 2500 I=1,K

          RRR=(XR(I,N)*YP1R(I,N1)-XI(I,N)*YP1I(I,N1))+
     1        (XXR(I,N)*YYP1R(I,N1)-XXI(I,N)*YYP1I(I,N1))
          RRI=(XI(I,N)*YP1R(I,N1)+XR(I,N)*YP1I(I,N1))+
     1        (XXI(I,N)*YYP1R(I,N1)+XXR(I,N)*YYP1I(I,N1))

          ER=RR-RRR
          EI=RI-RRI
          BER=BCONST*ER
          BEI=BCONST*EI

          BESCR=BER*XR(I,N)+BEI*XI(I,N)
          BESCI=BEI*XR(I,N)-BER*XI(I,N)
          BBESCR=BER*XXR(I,N)+BEI*XXI(I,N)
          BBESCI=BEI*XXR(I,N)-BER*XXI(I,N)

          YESR=YP1R(I,N1)+BESCR
          YESI=YP1I(I,N1)+BESCI
          YYESR=YYP1R(I,N1)+BBESCR
          YYESI=YYP1I(I,N1)+BBESCI

          EER=YESR-YP1R(I,N1)
          EEI=YESI-YP1I(I,N1)
          EE2R=YYESR-YYP1R(I,N1)
          EE2I=YYESI-YYP1I(I,N1)

          XGRR(I,N)=XGRR(I,N1)+((1.0-THETA)**2)*EER
          XGRI(I,N)=XGRI(I,N1)+((1.0-THETA)**2)*EEI
          XGR2R(I,N)=XGR2R(I,N1)+((1.0-THETA)**2)*EE2R
          XGR2I(I,N)=XGR2I(I,N1)+((1.0-THETA)**2)*EE2I

      YP1R(I,N)=YP1R(I,N1)+XGRR(I,N)+(1.0-(THETA**2))*EER
      YP1I(I,N)=YP1I(I,N1)+XGRI(I,N)+(1.0-(THETA**2))*EEI
      YYP1R(I,N)=YYP1R(I,N1)+XGR2R(I,N)+(1.0-(THETA**2))*EE2R
      YYP1I(I,N)=YYP1I(I,N1)+XGR2I(I,N)+(1.0-(THETA**2))*EE2I
```

```
2500  CONTINUE

      PYR=YR
      PYI=YI
      PYYR=YYR
      PYYI=YYI


3000       CONTINUE


4000             CONTINUE

C     CALCULATION OF ERROR RATES

      ER1SYM=IER1SYM/(COUNT-NTR)
      ER1BIT=(IER1+IER2)/(2*(COUNT-NTR))
      ER1DBIT=(IDER1+IDER2)/(2*(COUNT-NTR))

      ER2SYM=IER2SYM/(COUNT-NTR)
      ER2BIT=(IER3+IER4)/(2*(COUNT-NTR))
      ER2DBIT=(IDER3+IDER4)/(2*(COUNT-NTR))

      AVERSYM=(ER1SYM+ER2SYM)/2.0
      AVERBIT=(ER1BIT+ER2BIT)/2.0
      AVERDBIT=(ER1DBIT+ER2DBIT)/2.0


      EXQR=EQR/COUNT
      EXQI=EQI/COUNT
      VARQR=VQR/COUNT
      VARQI=VQI/COUNT
      EXQQR=EQQR/COUNT
      EXQQI=EQQI/COUNT
      VARQQR=VQQR/COUNT
      VARQQI=VQQI/COUNT

      SNRCALC=10.0*LOG10(1.0/SIGSQN)


C     PRINTING IMPORTANT RESULTS

      PRINT *,"BCONST=",BCONST
      PRINT *,"THETA=",THETA
      PRINT *,"PSI=",PSI

      WRITE(0,4010)NOD,ICOUNT,COUNT,NTR,IRETR
4010  FORMAT("NOD=",I7/"ICOUNT=",I7/'COUNT=",F12.2/"NTR=",I7/
     1 "IRETR=",I7)


      WRITE(0,4020)INFD,INDTN
4020  FORMAT("INFD=",I3/"INDTN=",I3/)

      WRITE(0,4030)EXQR,VARQR
4030  FORMAT("MEAN OF QR =",E17.10,2X,"VARIANCE OF QR =",E17.10)
      WRITE(0,4040)EXQI,VARQI
4040  FORMAT("MEAN OF QI =",E17.10,2X,"VARIANCE OF QI =",E17.10)
      WRITE(0,4050)EXQQR,VARQQR
4050  FORMAT("MEAN OF QQR=",E17.10,2X,"VARIANCE OF QQR=",E17.10)
      WRITE(0,4060)EXQQI,VARQQI
4060  FORMAT("MEAN OF QQI=",E17.10,2X,"VARIANCE OF QQI=",E17.10)
```

```
      WRITE(0,4070)SNRCALC
4070  FORMAT(/"SIGNAL-TO-NOISE RATIO =",F9.5,"DB'S"/)

      WRITE(0,4080)ER1SYM
4080  FORMAT("SYMBOL ERROR RATE NO.1        =",E17.10)
      WRITE(0,4090)ER1BIT
4090  FORMAT("BIT ERROR RATE NO.1           =",E17.10)
      WRITE(0,4100)ER1DBIT
4100  FORMAT("DIFFERENTIAL BIT ERROR RATE NO.1 =",E17.10/)

      WRITE(0,4110)ER2SYM
4110  FORMAT("SYMBOL ERROR RATE NO.2        =",E17.10)
      WRITE(0,4120)ER2BIT
4120  FORMAT("BIT ERROR RATE NO.2           =",E17.10)
      WRITE(0,4130)ER2DBIT
4130  FORMAT("DIFFERENTIAL BIT ERROR RATE NO.2 =",E17.10/)

      WRITE(0,4140)AVERSYM
4140  FORMAT("AVERAGE SYMBOL ERROR RATE     =",E17.10)
      WRITE(0,4150)AVERBIT
4150  FORMAT("AVERAGE BIT ERROR RATE        =",E17.10)
      WRITE(0,4160)AVERDBIT
4160  FORMAT("AVERAGE DIFFERENTIAL BIT ERROR  =",E17.10///)

      INDTN=INDTN+1

6000  CONTINUE



      STOP
      END
```

APPENDIX K

SIMULATION OF 4.8 KBIT/S HF RADIO MODEM AS GIVEN IN CHAPTER 7

```
/*JOB DGR1,EUELSJ2,ST=C20,C=C,TI=2500,
/* PW=SDMVP
PATTACH,LIBAPPL.
PATTACH,PROCLIB.
BEGIN,,GETFEP,FDIMP,FCE36,FO=BINARY.
SWITCH,FDIMP,RT=W.
FTN200,L=0,OPT.
LOAD,LIB=BLAS,NAGLIB.
GO.
####S
      PROGRAM FADFIL
      PARAMETER(NIMP=22,N=81,K=8,NN=32,NTAP=50,NEST=NIMP)
      PARAMETER(LLOUT=99)
      PARAMETER(MV1=1,MV2=1,MV3=2,MVTOT=MV1+MV2+MV3)
      INTEGER ILK1(32),ILK2(32),ILK3(32)
      INTEGER ILKP1(16,2),ILKP2(16,2),ILKP3(16,2)
      INTEGER IBIT1(N),IBIT2(N),IDBIT1(N),IDBIT2(N),ISR(N),ISI(N)
      INTEGER IXR(K,NN),IXI(K,NN),ITR(K,NN),ITI(K,NN)
      INTEGER INN(MVTOT)
      REAL HFR(NIMP),HFI(NIMP),WFR(NIMP),WFI(NIMP),YR(NIMP),YI(NIMP)
      REAL SR(N),SI(N)
      REAL XR(K,NN),XI(K,NN),GR(K),GI(K)
      REAL VR(K),VI(K),VVR(K),VVI(K)
      REAL COST(K),CT1(K),CT2(MVTOT),CT3(K)
      REAL SYR(NIMP),SYI(NIMP),SSYR(NIMP),SSYI(NIMP)
      REAL YFR(NTAP+1),YFI(NTAP+1),SYFR(NTAP),SYFI(NTAP)
      REAL RR(NTAP),RI(NTAP)
      REAL YRSR(NTAP,NIMP),YRSI(NTAP,NIMP)
      REAL YP1R(NEST),YP1I(NEST),GRDR(NEST),GRDI(NEST)
      REAL UPESR(NEST),UPESI(NEST),ERUPR(NEST),ERUPI(NEST)

C     VALUES TO VARIABLES AND ARRAYS

      IQ=21
      B=0.12
      THETA=0.95
      NOSNR=12
      KLOOP=57600
      SNR=14.0
      SNRD=0.5
      JPRD=0
      NPRED=JPRD+1
      NTR=1000
      IRTRN=1000
      RTRN=REAL(IRTRN)
      FACT=KLOOP/RTRN
      IFACT=INT(FACT)
      JNTR=NN1*(IFACT-1)
      SNR=SNR-SNRD
      N1=N-1
      MMVTOT=MVTOT
      MMV2=MV2
      MMV3=MV3
      NTAPM1=NTAP-1
      NTAPM2=NTAP-2
      NN1=NN-1
```

```
      DUM1=(1.0-THETA)*(1.0-THETA)
      DUM2=1.0-(THETA*THETA)

      DATA WFR/-0.02805,-0.23081,-0.50758,-0.49800,
     1        -0.16221, 0.11289, 0.10851,-0.00731,
     2        -0.04869,-0.00976, 0.01514, 0.00523,
     3        -0.00449, 0.00063, 0.00107,-0.00093,
     4         0.00000, 0.00000, 0.00000, 0.00000,
     5         0.00000, 0.00000/

      DATA WFI/ 0.01968, 0.16746, 0.39481, 0.40568,
     1         0.10503,-0.13375,-0.07360, 0.04669,
     2         0.02741,-0.01851,-0.00698, 0.01100,
     3         0.00286,-0.00221, 0.00136,-0.00045,
     4         0.00000, 0.00000, 0.00000, 0.00000,
     5         0.00000, 0.00000/

      DATA ILK1/0,0,0,1,1,0,1,1,0,1,1,1,0,0,1,0,
     1         1,0,0,0,1,1,0,1,1,1,1,0,0,1,0,0/

      DATA ILK2/1,1,-1,1,1,-1,-1,-1,-1,1,-1,-1,1,1,1,-1,
     1         1,-1,1,1,-1,-1,-1,1,-1,-1,1,-1,-1,1,1,1/

      DATA ILK3/0,0,0,1,1,0,1,1,1,0,0,0,1,1,0,1,
     1         0,1,1,1,0,0,1,0,1,1,1,0,0,1,0,0/

      KLK1=0
      DO 30 I=1,16
      DO 25 J=1,2
      ILKP1(I,J)=ILK1(KLK1+J)
      ILKP2(I,J)=ILK2(KLK1+J)
      ILKP3(I,J)=ILK3(KLK1+J)
   25  CONTINUE
      KLK1=KLK1+2
   30  CONTINUE


C     OPEN ALL RELEVANT FILES

      OPEN(18,FILE='FDIMP',FORM='UNFORMATTED',IOSTAT=IOSS)
      OPEN(LLOUT,FILE='OUTPUT',IOSTAT=JOSS)
      WRITE(LLOUT,71)IOSS,JOSS
   71  FORMAT(/' ERRORS IN EXECUTION OF THE OPEN STATEMENT=',I3,I3/
     1       ' '/)


      CALL G05CBF(IQ)


C     PREPARE FOR TRANSMISSION BY FIXING AN SNR VALUE

      DO 9000 IVAL=1,NOSNR

      SNR=SNR+SNRD
      STDVN=10.0**(-SNR/20.0)

      XX=0.0
      ICOUNT=0
      TPRER=0.0

      IERSYM=0
      IER1=0
      IER2=0
```

```
      IDER1=0
      IDER2=0

      IPDB1=0
      IPDB2=0


C     INITIALISING SIGNAL, AND NOISE  ARRAYS

      DO 100 I=1,N
      IBIT1(I)=0
      IBIT2(I)=0
      IDBIT1(I)=0
      IDBIT2(I)=0
      SR(I)=1.0
      SI(I)=1.0
      ISR(I)=1
      ISI(I)=1
100   CONTINUE

      DO 105 I=1,NTAP
      RR(I)=1.0
      RI(I)=1.0
105   CONTINUE

      DO 110 I=1,NIMP
      HFR(I)=0.0
      HFI(I)=0.0
110   CONTINUE


C     INITIALISING STORED VECTORS AND COSTS

      DO 130 I=1,K
      COST(I)=10000.0
      DO 120 J=1,NN
      XR(I,J)=1.0
      XI(I,J)=1.0
      IXR(I,J)=1
      IXI(I,J)=1
120   CONTINUE
130   CONTINUE
      COST(1)=0.0


C     INITIALISING ARRAY WHERE SIR WILL BE HELD

      DO 140 I=1,NTAP
      DO 135 J=1,NIMP
      YRSR(I,J)=0.0
      YRSI(I,J)=0.0
135   CONTINUE
      YRSR(I,1)=1.0
      YRSR(I,2)=-0.1
      YRSI(I,1)=0.0
140   CONTINUE


C     INITIALISING ARRAY WHERE PREDICTIONS ARE HELD

      DO 145 I=1,NEST
      YP1R(I)=0.0
      YP1I(I)=0.0
      GRDR(I)=0.0
```

```
      GRDI(I)=0.0
145   CONTINUE
      YP1R(1)=1.0
      YP1R(2)=-0.1


C     ENTERING MAIN LOOP

      DO 8000 IMAIN=1,KLOOP

      ICOUNT=ICOUNT+1
      COUNT=REAL(ICOUNT)

C     SHIFTING SIGNAL REGISTERS FOR INCOMING DATA SIGNAL

      DO 150 I=1,N1
      JX1=I+1
      IBIT1(I)=IBIT1(JX1)
      IBIT2(I)=IBIT2(JX1)
      IDBIT1(I)=IDBIT1(JX1)
      IDBIT2(I)=IDBIT2(JX1)
      SR(I)=SR(JX1)
      SI(I)=SI(JX1)
      ISR(I)=ISR(JX1)
      ISI(I)=ISI(JX1)
150   CONTINUE

      DO 155 I=1,NTAPM1
      JFX1=I+1
      RR(I)=RR(JFX1)
      RI(I)=RI(JFX1)
155   CONTINUE

C     SHIFTING STORED VECTORS TO ACCOMMODATE NEW VALUES

      DO 170 I=1,K
      DO 160 J=1,NN1
      JX2=J+1
      XR(I,J)=XR(I,JX2)
      XI(I,J)=XI(I,JX2)
      IXR(I,J)=IXR(I,JX2)
      IXI(I,J)=IXI(I,JX2)
160   CONTINUE
170   CONTINUE

C     SHIFTING SIR ARRAY IN PREPARATION FOR CURRENT SIR

      DO 175 I=1,NTAPM1
      DO 174 J=1,NIMP
      JFX2=I+1
      YRSR(I,J)=YRSR(JFX2,J)
      YRSI(I,J)=YRSI(JFX2,J)
174   CONTINUE
175   CONTINUE


C     INPUT SAMPLED IMPULSE RESPONSE

      READ(18)(YRSR(NTAP,I),I=1,NIMP)
      READ(18)(YRSI(NTAP,I),I=1,NIMP)
```

```
      IF (ICOUNT.EQ.NTAP.OR.MOD(ICOUNT,IRTRN).EQ.0) THEN
      DO 176 I=1,NIMP
      YP1R(I)=YRSR(1,I)
      YP1I(I)=YRSI(1,I)
      GRDR(I)=0.0
      GRDI(I)=0.0
  176   CONTINUE
      DO 178 I=1,K
      DO 177 J=1,NN1
      XR(I,J)=SR(J)
      XI(I,J)=SI(J)
      IXR(I,J)=ISR(J)
      IXI(I,J)=ISI(J)
  177   CONTINUE
      COST(I)=10000.0
  178   CONTINUE
      COST(1)=0.0
      END IF

      DO 179 I=1,NEST
      YR(I)=YP1R(I)
      YI(I)=YP1I(I)
  179   CONTINUE


C     CALCULATION OF PREDICTION ERRORS

      IF (ICOUNT.GT.NTR) THEN
      YEROR=0.0
      DO 180 I=1,NEST
      PRER1=(YP1R(I)-YRSR(1,I))**2
      PRER2=(YP1I(I)-YRSI(1,I))**2
      YEROR=YEROR+PRER1+PRER2
  180   CONTINUE
      TPRER=TPRER+YEROR
      END IF


C     CALL THE FILTER SUBROUTINE WHICH GIVES AN ESTIMATE OF THE CURRENT
C     MINIMUM PHASED CHANNEL AND THE CORRESPONDING FILTER COEFFICIENTS

      CALL DFILT(YR,YI,ICOUNT,SYR,SYI,YFR,YFI)


C     OBTAIN SCALED VERSION OF MINIMUM PHASED CHANNEL AND APPROPRIATELY
C     SCALED VERSION OF THE FILTER TAPS. THESE WILL BE HELD IN ARRAYS
C     SSYR,SSYI,SYFR AND SYFI, RESPECTIVELY.

      SYSQQ=SYR(1)*SYR(1)+SYI(1)*SYI(1)
      DO 183 J=1,NIMP
      SSYR(J)=(SYR(1)*SYR(J)+SYI(1)*SYI(J))/SYSQQ
      SSYI(J)=(SYR(1)*SYI(J)-SYI(1)*SYR(J))/SYSQQ
  183   CONTINUE

      DO 185 I=1,NTAP
      SYFR(I)=(SYR(1)*YFR(I)+SYI(1)*YFI(I))/SYSQQ
      SYFI(I)=(SYR(1)*YFI(I)-SYI(1)*YFR(I))/SYSQQ
  185   CONTINUE


C     CALCULATING THE INTERSYMBOL INTERFERENCE

      DO 200 I=1,K
      GR(I)=0.0
```

```
      GI(I)=0.0
      DO 195 J=2,NIMP
      JX3=NN+1-J
      GR(I)=GR(I)+XR(I,JX3)*SSYR(J)-XI(I,JX3)*SSYI(J)
      GI(I)=GI(I)+XR(I,JX3)*SSYI(J)+XI(I,JX3)*SSYR(J)
195   CONTINUE
200   CONTINUE


C     GENERATING SIGNAL IN BIT FORM

      XX=G05CAF(XX)
      IF(XX-0.5)210,210,220
210   IBIT1(N)=0
      GO TO 230
220   IBIT1(N)=1
230   XX=G05CAF(XX)
      IF(XX-0.5)240,240,250
240   IBIT2(N)=0
      GO TO 260
250   IBIT2(N)=1
260   CONTINUE


C     DIFFERENTIAL ENCODING AND GENERATION OF QPSK SIGNAL

      JEN1=IDBIT1(N1)*8+IDBIT2(N1)*4+IBIT1(N)*2+IBIT2(N)+1

      IDBIT1(N)=ILKP1(JEN1,1)
      IDBIT2(N)=ILKP1(JEN1,2)

      ISR(N)=ILKP2(JEN1,1)
      ISI(N)=ILKP2(JEN1,2)
      SR(N)=REAL(ISR(N))
      SI(N)=REAL(ISI(N))


C     GENERATION OF NOISE

      DO 350 LNM=1,2
      DO 330 I=1,NIMP-1
      JX4=I+1
      HFR(I)=HFR(JX4)
      HFI(I)=HFI(JX4)
330   CONTINUE
      HFR(NIMP)=G05DDF(0.0,STDVN)
      HFI(NIMP)=G05DDF(0.0,STDVN)
      WR=0.0
      WI=0.0
      DO 340 I=1,NIMP
      JX5=NIMP-I+1
      WR=WR+HFR(JX5)*WFR(I)-HFI(JX5)*WFI(I)
      WI=WI+HFR(JX5)*WFI(I)+HFI(JX5)*WFR(I)
340   CONTINUE
350   CONTINUE


C     CALCULATION OF RECEIVED SIGNAL

      RR(NTAP)=0.0
      RI(NTAP)=0.0
      DO 400 I=1,NIMP
      JX6=N+1-I
      RR(NTAP)=RR(NTAP)+SR(JX6)*YRSR(NTAP,I)-SI(JX6)*YRSI(NTAP,I)
```

```
      RI(NTAP)=RI(NTAP)+SR(JX6)*YRSI(NTAP,I)+SI(JX6)*YRSR(NTAP,I)
400   CONTINUE

      RR(NTAP)=RR(NTAP)+WR
      RI(NTAP)=RI(NTAP)+WI


C     PASSING RECEIVED SIGNAL THROUGH ADAPTIVE FILTER

      VFR=0.0
      VFI=0.0
      DO 450 I=1,NTAP
      JFX3=NTAP-I+1
      VFR=VFR+SYFR(I)*RR(JFX3)-SYFI(I)*RI(JFX3)
      VFI=VFI+SYFR(I)*RI(JFX3)+SYFI(I)*RR(JFX3)
450   CONTINUE


C     DETECTION BEGINS
C     THRESHOLD DETECTION TO OBTAIN LOWEST COST VECTOR.
C     CALCULATION OF THE COSTS OF THESE EXPANDED VECTORS.

      DO 500 I=1,K

      V1=VFR-GR(I)
      V2=VFI-GI(I)
      VR(I)=V1
      VI(I)=V2

      IF (V1.GE.0) THEN
      XR(I,NN)=1.0
      IXR(I,NN)=1
      ELSE
      XR(I,NN)=-1.0
      IXR(I,NN)=-1
      END IF

      IF (V2.GE.0) THEN
      XI(I,NN)=1.0
      IXI(I,NN)=1
      ELSE
      XI(I,NN)=-1.0
      IXI(I,NN)=-1
      END IF

      DR=V1-XR(I,NN)
      DI=V2-XI(I,NN)
      VVR(I)=DR
      VVI(I)=DI
      CT1(I)=COST(I)+DR*DR+DI*DI

500   CONTINUE


C     SELECT THE BEST VECTOR FROM THE 1K EXPANDED VECTORS

      CC=1000000.0
      DO 650 I=1,K
      IF(CT1(I)-CC)630,650,650
630   CC=CT1(I)
      LC1=I
650   CONTINUE
      CT2(1)=CC
      INN(1)=LC1
```

```
      CT1(LC1)=10000000.0

C     OBTAINING BIT VALUE OF DETECTED SIGNAL

      KADD=IXR(LC1,1)+IXI(LC1,1)

      IF(KADD-0)670,690,680
670   IDB1=1
      IDB2=1
      GO TO 700
680   IDB1=0
      IDB2=0
      GO TO 700
690   IF(IXR(LC1,1).EQ.1)THEN
      IDB1=1
      IDB2=0
      ELSE
      IDB1=0
      IDB2=1
      END IF
700   CONTINUE

C     DIFFERENTIAL DECODING OPERATION

      JEN2=IPDB1*8+IPDB2*4+IDB1*2+IDB2+1

      ISB1=ILKP3(JEN2,1)
      ISB2=ILKP3(JEN2,2)

C     ERROR COUNT

      IF(ICOUNT.LE.NTR)GO TO 800
      IF(ISR(1)-IXR(LC1,1))780,770,780
770   IF(ISI(1)-IXI(LC1,1))780,790,780
780   IERSYM=IERSYM+1
790   IF(IDBIT1(1).NE.IDB1)IER1=IER1+1
      IF(IDBIT2(1).NE.IDB2)IER2=IER2+1
      IF(IBIT1(1).NE.ISB1)IDER1=IDER1+1
      IF(IBIT2(1).NE.ISB2)IDER2=IDER2+1
800   CONTINUE

C     DISCARDING THE VECTORS WHICH DO NOT AGREE

      DO 830 I=1,K
      IF(IXR(I,1)-IXR(LC1,1))820,810,820
810   IF(IXI(I,1)-IXI(LC1,1))820,830,820
820   CT1(I)=100000.0
830   CONTINUE

C     SELECTING (MVTOT-1) MORE VECTORS FOR THE CORRESPONDING EXPANSIONS

      IF (MMVTOT.EQ.1) GO TO 870

      DO 860 I=1,MMVTOT-1
      CC=1000000.0
         DO 850 J=1,K
         IF(CT1(J)-CC)840,850,850
840        CC=CT1(J)
         LC2=J
```

```
850       CONTINUE
      CT2(I+1)=CC
      INN(I+1)=LC2
      CT1(LC2)=10000000.0
860   CONTINUE

870   CONTINUE


C     TRANSFER AND EXPANSION OF THE BEST (MV1) VECTORS INTO
C     4 WAYS AND CALCULATING THEIR COSTS.

      MC1=1
      JD1=0
      DO 1000 I=1,MV1
      LC3=INN(I)
      CT3(MC1)=CT2(I)
      DO 950 IJ1=1,4
      JD1=JD1+1
            DO 925 J=1,NN
            ITR(JD1,J)=IXR(LC3,J)
            ITI(JD1,J)=IXI(LC3,J)
925           CONTINUE
950   CONTINUE

      ITR(JD1,NN)=-IXR(LC3,NN)
      ITI(JD1,NN)=-IXI(LC3,NN)

      ITI(JD1-1,NN)=-IXI(LC3,NN)

      ITR(JD1-2,NN)=-IXR(LC3,NN)

            JD2=2
            DO 975 IJ2=2,4
            JD3=JD1-JD2
            DDR=VR(LC3)-REAL(ITR(JD3,NN))
            DDI=VI(LC3)-REAL(ITI(JD3,NN))
            CT3(JD3)=COST(LC3)+DDR*DDR+DDI*DDI
            JD2=JD2-1
975           CONTINUE
      MC1=MC1+4
1000  CONTINUE


C     TRANSFER AND EXPANSION OF THE NEXT BEST (MV2) VECTORS
C     INTO 2 WAYS AND CALCULATING THEIR COSTS

      IF (MMV2.EQ.0) GO TO 1060

      ND1=MV1*4
      DO 1050 I=1,MMV2
      LC3=INN(MV1+I)
      CT3(MC1)=CT2(MV1+I)
            DO 1010 IJ1=1,2
            ND1=ND1+1
            DO 1005 J=1,NN
            ITR(ND1,J)=IXR(LC3,J)
            ITI(ND1,J)=IXI(LC3,J)
1005          CONTINUE
1010          CONTINUE

      TMXR=IXR(LC3,NN)
      TMXI=IXI(LC3,NN)
      VMR=SIGN(2.1,VVR(LC3))
```

```
      IVMR=INT(VMR)
      VMI=SIGN(2.1,VVI(LC3))
      IVMI=INT(VMI)

      IF (ABS(VVI(LC3))-ABS(VVR(LC3))) 1022,1022,1030

1022  ITR(ND1,NN)=TMXR+IVMR
      IF (ABS(ITR(ND1,NN))-2) 1045,1024,1024
1024  ITR(ND1,NN)=TMXR
      ITI(ND1,NN)=TMXI+IVMI
      IF (ABS(ITI(ND1,NN))-2) 1045,1026,1026
1026  ITI(ND1,NN)=TMXI-IVMI
      GO TO 1045

1030  ITI(ND1,NN)=TMXI+IVMI
      IF (ABS(ITI(ND1,NN))-2) 1045,1032,1032
1032  ITR(ND1,NN)=TMXR+IVMR
      ITI(ND1,NN)=TMXI
      IF (ABS(ITR(ND1,NN))-2) 1045,1034,1034
1034  ITR(ND1,NN)=TMXR-IVMR

1045  CONTINUE

      DDR=VR(LC3)-REAL(ITR(ND1,NN))
      DDI=VI(LC3)-REAL(ITI(ND1,NN))
      CT3(ND1)=COST(LC3)+DDR*DDR+DDI*DDI

      MC1=MC1+2
1050  CONTINUE

1060  CONTINUE

C     TRANSFER OF NEXT BEST (MV3) VECTORS EXPANDED SINGLY

      IF (MMV3.EQ.0) GO TO 1068

      DO 1066 I=1,MMV3
      LC3=INN(MV1+MV2+I)
      DO 1064 J=1,NN
      ITR(MC1,J)=IXR(LC3,J)
      ITI(MC1,J)=IXI(LC3,J)
1064  CONTINUE
      CT3(MC1)=CT2(MV1+MV2+I)
      MC1=MC1+1
1066  CONTINUE

1068  CONTINUE

C     TRANSFERRING THE FINAL K VECTORS AND COSTS TO THEIR ORIGINAL
C     STORE, READY FOR THE NEXT DETECTION PROCESS.

      DO 1080 I=1,K
      DO 1070 J=1,NN
      IXR(I,J)=ITR(I,J)
      IXI(I,J)=ITI(I,J)
      XR(I,J)=REAL(IXR(I,J))
      XI(I,J)=REAL(IXI(I,J))
1070  CONTINUE
      COST(I)=CT3(I)
1080  CONTINUE
```

```
C     SUBTRACT COSTS TO BRING THE SMALLEST COST TO ZERO

      DO 1090 I=2,K
      COST(I)=COST(I)-COST(1)
1090  CONTINUE
      COST(1)=0.0


      IPDB1=IDB1
      IPDB2=IDB2

C     ESTIMATION BEGINS
C     FORMING ESTIMATE OF THE RECEIVED SIGNAL AND CALCULATING THE
C     ERROR IN THAT ESTIMATE

      RER=0.0
      REI=0.0
      DO 1110 I=1,NEST
      KX1=NN+1-I
      RER=RER+XR(1,KX1)*YP1R(I)-XI(1,KX1)*YP1I(I)
      REI=REI+XR(1,KX1)*YP1I(I)+XI(1,KX1)*YP1R(I)
1110  CONTINUE

      ERESR=RR(1)-RER
      ERESI=RI(1)-REI


C     OBTAINING THE UPDATED ESTIMATE AND ITS ERROR, WHERE THE
C     LATTER IS OBTAINED FIRST, IN ORDER TO MAKE THE SOFTWARE
C     MORE EFFIFCIENT.

      DO 1120 I=1,NEST
      KX2=NN+1-I
      ERUPR(I)=B*(ERESR*XR(1,KX2)+ERESI*XI(1,KX2))
      ERUPI(I)=B*(ERESI*XR(1,KX2)-ERESR*XI(1,KX2))
      UPESR(I)=YP1R(I)+ERUPR(I)
      UPESI(I)=YP1I(I)+ERUPI(I)
1120  CONTINUE


C     PREDICTION PROCESS BEGINS

      DO 1130 I=1,NEST
      GRDR(I)=GRDR(I)+(DUM1*ERUPR(I))
      GRDI(I)=GRDI(I)+(DUM1*ERUPI(I))
      YP1R(I)=YP1R(I)+GRDR(I)+(DUM2*ERUPR(I))
      YP1I(I)=YP1I(I)+GRDI(I)+(DUM2*ERUPI(I))
1130  CONTINUE


8000  CONTINUE


C     CONFIRMATION OF SNR VALUE AND CALCULATION OF ERROR RATES

      SNRCALC=10.0*LOG10(1.0/(STDVN*STDVN))

      ERSYM=IERSYM/(COUNT-NTR-JNTR)
      ERBIT=(IER1+IER2)/(2.0*(COUNT-NTR-JNTR))
      ERDBIT=(IDER1+IDER2)/(2.0*(COUNT-NTR-JNTR))
```

```
      IF (TPRER.NE.0) THEN
      AVPRER=10.0*LOG10(TPRER/(COUNT-NTR))
      ELSE
      AVPRER=0.0
      END IF


C     PRINTING IMPORTANT RESULTS

      WRITE(LLOUT,8110)KLOOP,NTR,IQ,K
8110  FORMAT(' KLOOP=',I10/' NTR=',I10/' IQ=',I10/' K=',I10)
      WRITE(LLOUT,8120)ICOUNT,COUNT
8120  FORMAT(' ICOUNT=',I10/' COUNT=',F12.2)
      WRITE(LLOUT,8123)IFACT,JNTR
8123  FORMAT(' IFACT=',I10/' JNTR=',I10/)

      WRITE(LLOUT,8125)B,THETA
8125  FORMAT(' B=',F8.4/' THETA=',F8.4/)

      WRITE(LLOUT,8130)SNRCALC
8130  FORMAT(' THEORETICAL SIGNAL-TO-NOISE RATIO=',F9.3,' DB')

      WRITE(LLOUT,8135)AVPRER
8135  FORMAT(' AVERAGE ERROR IN ONE-STEP PREDICTION =',E20.10,' DB')

      WRITE(LLOUT,8140)ERSYM
8140  FORMAT(' SYMBOL ERROR RATE        =',E20.10)
      WRITE(LLOUT,8150)ERBIT
8150  FORMAT(' BIT ERROR RATE           =',E20.10)
      WRITE(LLOUT,8160)ERDBIT
8160  FORMAT(' DIFFERENTIAL BIT ERROR RATE =',E20.10//)

      REWIND(18)


9000  CONTINUE


      STOP
      END
      SUBROUTINE DFILT(YR,YI,ICOUNT,SYR,SYI,YFR,YFI)


      PARAMETER(NINRT=9,NIMP=22,NTAP=50)
      REAL YR(NIMP),YI(NIMP),SYR(NIMP),SYI(NIMP)
      REAL ER(NIMP+1),EI(NIMP+1)
      REAL FR(NIMP+1),FI(NIMP+1)
      REAL RREAL(NIMP),RIMAG(NIMP)
      REAL FSPR(NINRT),FSPI(NINRT),STPR(NIMP),STPI(NIMP)
      REAL YFR(NTAP+1),YFI(NTAP+1)
      REAL QR(NTAP+1),QI(NTAP+1)
      REAL PREAL(NIMP),PIMAG(NIMP)


C     VALUES TO VARIABLES AND ARRAYS

      C=1.0
      D=10.0E-10
      AD=1.05
      ALTHD=1.0/AD
      NROOT=0
      JREF=1
      NIMPP1=NIMP+1
```

```
      NIMPM2=NIMP-2
      NTAPP1=NTAP+1
      NDIVG=40

      DATA FSPR / 0.00000, 0.90909, 0.00000, 0.00000,-0.90909,
     1          0.64282, 0.64282,-0.64282,-0.64282 /

      DATA FSPI / 0.00000, 0.00000,-0.90909, 0.90909, 0.00000,
     1          -0.64282, 0.64282, 0.64282,-0.64282 /

      IFSPR=INT(FSPR(NINRT)*100.0)
      IFSPI=INT(FSPI(NINRT)*100.0)

      IF (ICOUNT.EQ.1) THEN
        DO 50 I=1,NINRT
        STPR(I)=FSPR(I)
        STPI(I)=FSPI(I)
   50     CONTINUE
        DO 60 I=NINRT+1,NIMP
        STPR(I)=0.0
        STPI(I)=0.0
   60     CONTINUE
      END IF

      DO 100 I=1,NIMP
      SYR(I)=YR(I)
      SYI(I)=YI(I)
      RREAL(I)=0.0
      RIMAG(I)=0.0
  100 CONTINUE

  200 CONTINUE


C     ONE TAP FEEDBACK FILTER

      BETAR=STPR(JREF)
      BETAI=STPI(JREF)

      DO 1000 I=1,NDIVG

        DO 510 J=1,NIMPP1
        ER(J)=0.0
        EI(J)=0.0
  510     CONTINUE

      PR=0.0
      PI=0.0
        DO 520 J=1,NIMP
        J1=NIMP+1-J
        FOR=SYR(J1)-(BETAR*PR-BETAI*PI)
        FOI=SYI(J1)-(BETAR*PI+BETAI*PR)
        PR=FOR
        PI=FOI
        ER(J1)=FOR
        EI(J1)=FOI
  520     CONTINUE

C     ESTIMATE VALUE OF EPSILON

      ALFAR=-BETAR
      ALFAI=-BETAI
      EPSR=ER(NIMP)
```

```
      EPSI=EI(NIMP)
        DO 530 J=1,NIMPM2
        J2=NIMP-J
        PALFAR=EPSR*ALFAR-EPSI*ALFAI
        PALFAI=EPSR*ALFAI+EPSI*ALFAR
        EPSR=PALFAR+ER(J2)
        EPSI=PALFAI+EI(J2)
530       CONTINUE

C     OBTAIN CONDITION FOR CONVERGENCE

      EPSMAG=EPSR*EPSR+EPSI*EPSI
      DELR=(ER(1)*EPSR+EI(1)*EPSI)/EPSMAG
      DELI=(EI(1)*EPSR-ER(1)*EPSI)/EPSMAG
      DELMAG=DELR*DELR+DELI*DELI

C     AS LONG AS CONVERGENCE IS NOT ACHEIVED OR ALGORITHM HAS
C     NOT DIVERGED, DO ITERATION AGAIN.

      IF (DELMAG.LT.D) THEN
      GO TO 2050
      ELSE
      DELR=C*DELR
      DELI=C*DELI
      BETAR=BETAR+DELR
      BETAI=BETAI+DELI

      DL1=SQRT(BETAR*BETAR+BETAI*BETAI)
        IF (DL1.GT.ALTHD) THEN
        GO TO 1050
        END IF
      END IF

1000  CONTINUE


1050  CONTINUE

C     FIND NEW STARTING POINT AFTER DIVERGENCE

      IF (ICOUNT.EQ.1) THEN
        IF (JREF.LT.NINRT) THEN
        JREF=JREF+1
        GO TO 200
        ELSE
        GO TO 2200
        END IF
      ELSE
      ISTPR=INT(STPR(JREF)*100.0)
      ISTPI=INT(STPI(JREF)*100.0)
        IF (ISTPR.EQ.IFSPR.AND.ISTPI.EQ.IFSPI) THEN
        GO TO 2200
        ELSE
        JREF=JREF+1
        GO TO 200
        END IF
      END IF


2050  CONTINUE

C     ALTER CHANNEL WITH OBTAINED ROOT BY PASSING THROUGH
C     TWO TAP FEEDFORWARD FILTER
```

```
      NROOT=NROOT+1
      RREAL(NROOT)=BETAR
      RIMAG(NROOT)=BETAI
      FR(1)=ER(1)
      FI(1)=EI(1)
      DO 540 J=2,NIMPP1
      J3=J-1
      FR(J)=ER(J)+ER(J3)*BETAR+EI(J3)*BETAI
      FI(J)=EI(J)+EI(J3)*BETAR-ER(J3)*BETAI
      SYR(J3)=FR(J)
      SYI(J3)=FI(J)
  540 CONTINUE

C     FIND NEW STARTING POINT AFTER CONVERGENCE

      IF (ICOUNT.EQ.1) THEN
        IF (JREF.LT.NINRT) THEN
        JREF=JREF+1
        GO TO 200
        ELSE
        GO TO 2200
        END IF
      ELSE
        IF (JREF.LE.NPROOT) THEN
        JREF=JREF+1
        GO TO 200
        ELSE
        GO TO 200
        END IF
      END IF

 2200 CONTINUE

C     SETTING UP STARTING POINTS FOR NEXT TIME INSTANT WHEN ALL
C     ROOTS HAVE BEEN FOUND.

      NTOT1=NROOT+NINRT+1
      NPROOT=NROOT
      DO 2250 J=1,NROOT
      STPR(J)=RREAL(J)
      STPI(J)=RIMAG(J)
 2250 CONTINUE
      DO 2260 J=1,NINRT
      STPR(NROOT+J)=FSPR(J)
      STPI(NROOT+J)=FSPI(J)
 2260 CONTINUE
      IF (NTOT1.LE.NIMP) THEN
      DO 2270 J=NTOT1,NIMP
      STPR(J)=0.0
      STPI(J)=0.0
 2270 CONTINUE
      ELSE
      GO TO 3000
      END IF


 3000 CONTINUE

      DO 3005 J=1,NROOT
      CETAR=RREAL(J)
      CETAI=RIMAG(J)
```

```
      RT=CETAR*CETAR+CETAI*CETAI
      PREAL(J)=(-1.0*CETAR)/RT
      PIMAG(J)=CETAI/RT
3005  CONTINUE

C     AT THIS POINT SYR AND SYI CONTAIN THE MINIMUM PHASED VERSION
C     OF YR AND YI FOR THE PRESENT TIME INSTANT.

C     START CALCULATION OF THE TAP GAINS OF THE FILTER.

C     INITIALIZE TAP GAINS

      DO 4500 J=1,NTAPP1
      YFR(J)=0.0
      YFI(J)=0.0
4500  CONTINUE
      YFR(NTAP)=1.0

C     ADJUST THE TAP GAINS OF THE FILTER BY PASSING EACH CONVERGED
C     ROOT IN ACCORDANCE WITH THE GIVEN ALGORITHM.

      DO 5000 IJK=1,NROOT

      GAMAR=RREAL(IJK)
      GAMAI=RIMAG(IJK)

      DO 4610 J=1,NTAPP1
      QR(J)=0.0
      QI(J)=0.0
4610  CONTINUE

C     FEEDING THROUGH TWO TAP FEEDFORWARD FILTER

      QR(1)=YFR(1)
      QI(1)=YFI(1)
      DO 4620 J=1,NTAP
      J4=J+1
      QR(J4)=YFR(J4)+(YFR(J)*GAMAR+YFI(J)*GAMAI)
      QI(J4)=YFI(J4)+(YFI(J)*GAMAR-YFR(J)*GAMAI)
4620  CONTINUE

C     FEEDING OUTPUT FROM ABOVE FILTER THROUGH A ONE TAP FEEDBACK
C     FILTER. THE SEQUENCE IS FED IN REVERSE ORDER. THE OUTPUT
C     FROM THIS ONE TAP FILTER IS ALSO OBTAINED IN REVERSE ORDER.

      QGR=0.0
      QGI=0.0
      DO 4630 J=1,NTAP
      J5=NTAP+2-J
      QFR=QR(J5)-(GAMAR*QGR-GAMAI*QGI)
      QFI=QI(J5)-(GAMAR*QGI+GAMAI*QGR)
      QGR=QFR
      QGI=QFI
      YFR(J5-1)=QFR
      YFI(J5-1)=QFI
4630  CONTINUE
```

```
5000  CONTINUE


      RETURN
      END
####S
```