ADAPTIVE EQUALIZATION OF FADING RADIO CHANNELS

by

Parveen Kumar Shukla, B.A., M.Sc.

A thesis submitted for the degree of

Doctor of Philosophy

in the Faculty of Engineering of the University of London,

and for the Diploma of Imperial College.

Digital Communications Section
Department of Electrical Engineering
Imperial College of Science, Technology and Medicine
Exhibition Road
London SW7 2BT

December 1988

Abstract

The primary objective of this thesis is to investigate the transmission of 4, 16 and 64 QAM data signals at 2.4 kbaud s^{-1} over a voiceband HF radio link; the data rate respectively being 4.8, 9.6 and 14.4 kbit s^{-1}. The particular problems associated with the channel medium are that it is a multipath channel, thereby giving rise to intersymbol interference (ISI), which can be quite severe at times, and also that its characteristic changes significantly with time. Two ISI mitigating devices, or equalizers, are examined; namely, the decision feedback equalizer (DFE), and the maximum likelihood sequence estimator (MLSE), or an approximation to it (near-MLSE). Their performance is compared using computer simulated models of two-path and three-path HF channels which have additive gaussian noise. The time-varying nature of the channel requires that the receiver be made adaptive, and the performances offered by the steepest descent (SD) and recursive least squares (RLS) tracking algorithms, both of which are decision-directed, is investigated. An important question that is addressed is, what steps need to be taken to produce acceptable system performance at higher data rates?

It is shown that the superiority of the more complex MLSE/near-MLSE over the DFE is not too great, and that this advantage is only due to error propagation in the latter. In the absence of any special arrangement (e.g. periodic training) by which the receiver can re-align itself, 16 QAM appears to be the highest feasible signal constellation size. With the periodic insertion of known symbols into the data stream, it is shown that performance can be maintained near the level for which decision errors do not affect the tracking algorithm; with the facility of a request-for-training (RTS) link to the transmitter, it is shown that performance can be significantly improved for all constellation sizes, with only a small loss in the useful data rate. The investigations have also produced some interesting by-products. A superior way of implementing the adaptive DFE than with the conventional RLS approach is presented. The method relies on the process of channel estimation, which is theoretically and experimentally analysed for both the SD and RLS algorithms. It is demonstrated that the simpler SD algorithm gives a comparable level of accuracy with the more complex RLS algorithm. Using this new approach, it is also shown how simple predictions of the performance degradation of the DFE due to tracking error can be made. Finally, it is observed that a particular form of RLS algorithm is unstable under certain conditions.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. L. F. Turner.

I am glad to have known some friendly and helpful people during my time at Imperial College.

Special thanks go to my family for their patience and support.

CONTENTS

Chapter 1 Page

INTRODUCTION 12

1.1 Characteristics of the HF Channel 13

1.2 Receiver Designs 19

1.2.1 Receivers for Channels with Negligible ISI 21

1.2.2 Receivers for Channels with ISI 25

1.3 Objectives and Outline of the Thesis 27

1.3.1 Related Work 27

1.3.2 Objectives of the Thesis 29

1.3.3 Brief Outline of Chapters 30

Chapter 2

DETECTION TECHNIQUES IN A QAM TRANSMISSION SYSTEM 32

2.1 The QAM Transmission System 32

2.2 Equalization Strategies at the Receiver 38

2.2.1 Linear Equalizer (LE) 43

2.2.2 Decision Feedback Equalizer (DFE) 44

2.2.3 Matched Filter Equalizer (MFE) 46

2.3 Optimum Detection and the MLSE 47

2.3.1 Implementation of the MLSE by the VA 49

2.3.2 Lower Bound to Performance 55

2.4 Comparison 56

2.5 Fractionally-Spaced Transversal Filters 58

2.6 Effect of Time-Varying Channel 63

2.6.1 MMSE DFE with Finite Number of Taps 64

2.6.2 MMSE MFE and ZF MFE 67

2.7 Other Detection Devices 69

Chapter 3 Page

TRACKING ALGORITHMS 71

3.1 The Steepest Descent (SD) Algorithm 71

3.2 Recursive Least Squares (RLS) Algorithms 75

3.3 Application to Equalizer Adaptation 80

3.3.1 The DFE(N,g) 80

3.3.2 Channel Estimation 80

3.3.3 Training/Decision-Directed Mode 81

Chapter 4

THE HF CHANNEL MODEL 83

4.1 The Gaussian-Scatter Model 83

4.2 Carrier Phase Recovery 86

4.3 Channel Configurations Tested 88

4.4 Notes on Implementation 91

4.4.1 Model of the Transmission System 91

4.4.2 Signal-to-Noise Ratio 93

4.4.3 Generation of $\{y_{ki}\}$, $\{n_k\}$ and $\{s_k\}$ 94

4.5 Time Span Observed 98

Chapter 5

MODE I PERFORMANCE 101

5.1 Performance Criteria 102

5.1.1 Probability of Error 102

5.1.1.1 Threshold Detection, as used in the DFE and MFE 102

5.1.1.2 MLSE/near-MLSE 104

5.1.2 Mean Square Error 104

5.2 ZF MFE(g+1) 105

5.3 MMSE DFE(N,g)-I$_1$ 108

Page

5.4 MLSE/near-MLSE and MMSE DFE(6,g) 110

Chapter 6

CHANNEL ESTIMATION 127

6.1 Channel Estimation using the RLS (SRK) Algorithm 127

6.2 Channel Estimation using the SD Algorithm 133

6.3 Start-up Convergence of SRK and SD Algorithms 140

6.4 Improving the Performance at High *SNR* 143

Chapter 7

MODE II PERFORMANCE 157

7.1 Computation of DFE Taps from Channel Estimate 157

7.2 Performance Comparison of CE and Conventional Methods 161

7.3 Revised SRK Algorithm of Hsu [16] 168

7.4 Performance of DFE(6,g)-II$_1$ 172

7.5 MLSE/near-MLSE and DFE(6,g) 175

Chapter 8

MODE III PERFORMANCE 197

8.1 Performance of DFE(6,g)-III 198

8.2 DFE-III with Periodic Training Sequence (PTS) 207

8.3 DFE-III with Request for Training Sequence (RTS) 210

Chapter 9

CONCLUSIONS 222

APPENDICES 228

A: 229 B: 231 C: 234 D: 236 E: 238 F: 244 G: 246 H: 249 I: 256

REFERENCES 307

List of Figures and Tables Page

Fig. 2.1 A QAM Transmission System 33

Fig. 2.2 An SSB-PAM Transmission System 37

Fig. 2.3 QAM Baseband Transmission System 39

Fig. 2.4 Digital implementation of T-spaced transversal filter and KT/M fractionally-spaced filter. 59

Fig. 2.5 Receiver arrangement for $f_y \leq \frac{1}{T}$ 61

Fig. 2.6 Decision Feedback Equalizer: DFE(N,g) 65

Fig. 2.7 Matched Filter Equalizer: MFE(g+1) 68

Fig. 4.1 Model of i^{th} path of HF channel 84

Fig. 4.2 Channel configurations 90

Fig. 4.3 Model of the Transmission System 92

Fig. 4.4 Approximation to Gaussian Function $\left(\sqrt{2.5\sqrt{\pi}}\right)^{-1}.e^{-t^2/2T_2^2}$ 97

Fig. 4.5 Fading Pattern for Channel A. 99

Fig. 4.6 Fading Pattern for Channel B. 100

Fig. 5.1 Mode I_1 Performance of ZF MFE(2) on Channel A. (P_e) 116

Fig. 5.2 Mode I_1 Performance of MFE(2) on Channel A. (ξ) 117

Fig. 5.3 Mode I_1 Performance of ZF MFE(3) on Channel B. (P_e) 118

Fig. 5.4 Mode I_1 Performance of MFE(3) on Channel B. (ξ) 119

Fig. 5.5 Mode I_1 Performance of DFE(N,1) on Channel A. (P_e) 120

Fig. 5.6 Mode I_1 Performance of DFE(N,1) on Channel A. (ξ) 121

Fig. 5.7 Mode I_1 Performance of DFE(N,2) on Channel B. (P_e) 122

Fig. 5.8 Mode I_1 Performance of DFE(N,2) on Channel B. (ξ) 123

Fig. 5.9 Ratio (dB) of Average Mean Square Noise to Average Mean Square ISI vs. SNR, in mode I_1. 124

Fig. 5.10 Mode I_1 Performance on Channel A. 125

Fig. 5.11 Mode I_1 Performance on Channel B. 126

Page

Fig. 6.1 SNR vs. optimum ω for RLS algorithm on Channel A(g=1) and Channel B(g=2). 146

Fig. 6.2 Theoretical and measured ϵ', ϵ/σ_s^2 vs. SNR for SRK algorithm, with optimum ω, on Channel A. 147

Fig. 6.3 Theoretical and measured ϵ', ϵ/σ_s^2 vs. SNR for SRK algorithm, with optimum ω, on Channel B. 148

Fig. 6.4 Variation of ϵ' with ω for SRK algorithm, f.r.=2.0 Hz. 149

Fig. 6.5 SNR vs. optimum μ' for SD algorithm on Channel A(g=1). 150

Fig. 6.6 SNR vs. optimum μ' for SD algorithm on Channel B(g=2). 151

Fig. 6.7 Theoretical and measured ϵ' vs. SNR for SD algorithm, with optimum μ', on Channel A. 152

Fig. 6.8 Theoretical and measured ϵ' vs. SNR for SD algorithm, with optimum μ', on Channel B. 153

Fig. 6.9 Variation of ϵ' with μ' for SD algorithm, f.r.=2.0 Hz. 154

Fig. 6.10 Start-up convergence of error in channel estimate for Channel A, f.r.=2.0 Hz. 155

Fig. 6.11 Start-up convergence of error in channel estimate for Channel B, f.r.=2.0 Hz. 156

Fig. 7.1 Start-up convergence of MSE for DFE(6,1) with 4 QAM on Channel A, SNR=22.5 dB, f.r.=1.0 Hz. 181

Fig. 7.2 Start-up convergence of MSE for DFE(6,1) with 64 QAM on Channel A, SNR=40.0 dB, f.r.=1.0 Hz. 182

Fig. 7.3 Start-up convergence of MSE for DFE(6,2) with 4 QAM on Channel B, SNR=22.5 dB, f.r.=1.0 Hz. 183

Fig. 7.4 Start-up convergence of MSE for DFE(6,2) with 64 QAM on Channel B, SNR=40.0 dB, f.r.=1.0 Hz. 184

Page

Fig. 7.5 Optimum ω vs. SNR at f.r.=1.0 Hz for conventional implementation of DFE-II$_1$. 185

Fig. 7.6 Performance comparison of CE method and conventional method of implementing DFE(6,1)-II$_1$ on Channel A, f.r.=1.0 Hz. (P_e) 186

Fig. 7.7 Performance comparison of CE method and conventional method of implementing DFE(6,1)-II$_1$ on Channel A, f.r.=1.0 Hz. (ξ) 187

Fig. 7.8 Performance comparison of CE method and conventional method of implementing DFE(6,2)-II$_1$ on Channel B, f.r.=1.0 Hz. (P_e) 188

Fig. 7.9 Performance comparison of CE method and conventional method of implementing DFE(6,2)-II$_1$ on Channel B, f.r.=1.0 Hz. (ξ) 189

Fig. 7.10 Plot of function $M{=}1{+}\ln(\omega^{-1}-1).(\ln\omega)^{-1}$ 190

Fig. 7.11 Performance of DFE(6,1)-I$_1$, and DFE(6,1)-II$_1$ at f.r.=1.0 Hz and 2.0 Hz, on Channel A. (P_e) 191

Fig. 7.12 Performance of DFE(6,1)-I$_1$, and DFE(6,1)-II$_1$ at f.r.=1.0 Hz and 2.0 Hz, on Channel A. (ξ) 192

Fig. 7.13 Performance of DFE(6,2)-I$_1$, and DFE(6,2)-II$_1$ at f.r.=1.0 Hz and 2.0 Hz, on Channel B. (P_e) 193

Fig. 7.14 Performance of DFE(6,2)-I$_1$, and DFE(6,2)-II$_1$ at f.r.=1.0 Hz and 2.0 Hz, on Channel B. (ξ) 194

Fig. 7.15 Mode I and mode II performance, at f.r.=1.0 Hz, on Channel A. 195

Fig. 7.16 Mode I and mode II performance, at f.r.=1.0 Hz, on Channel B. 196

Fig. 8.1 Mode II$_2$ and mode III performance of DFE(6,1) on Channel A, f.r.=1.0 Hz. 216

Fig. 8.2 Mode II$_2$ and mode III performance of DFE(6,2) on Channel B, f.r.=1.0 Hz. 217

Fig. 8.3 Performance of DFE(6,1)-II$_2$, and DFE(6,1)-III with 10% PTS, on Channel A, f.r.=1.0 Hz. 218

Page

Fig. 8.4 Performance of DFE(6,2)-II_2, and DFE(6,2)-III with 10% PTS, on Channel B, f.r.=1.0 Hz. 219

Fig. 8.5 Performance of DFE(6,1)-II_1, and DFE(6,1)-III with RTS, on Channel A, f.r.=1.0 Hz. 220

Fig. 8.6 Performance of DFE(6,2)-II_1, and DFE(6,2)-III with RTS, on Channel B, f.r.=1.0 Hz. 221

Table 2.1 Receiver designs for different equalizers. 56

Table 5.1 Advantage of ZF MFE over MMSE DFE. 109

Table 5.2 Error counts for mode I on Channel A. 112

Table 5.3 Error counts for mode I on Channel B. 113

Table 6.1 Comparison of convergence times for SRK and SD algorithms. 141

Table 7.1 Comparison of requirements for implementing adaptive DFE by way of CE method and conventional method. 160

Table 7.2 Comparison of convergence times for CE and conventional methods. 162

Table 7.3 *SNR* advantage (dB) of CE method over conventional method, using P_e and ξ criteria. 164

Table 7.4 Error count performance figures for revised SRK algorithm of [16] on Channel B with 64 QAM. 170

Table 7.5 *SNR* degradation (dB) from mode I_1 to mode II_1. 173

Table 7.6 Error counts for mode II on Channel A, using SRK algorithm. 176

Table 7.7 Error counts for mode II on Channel B, using SRK algorithm. 177

Table 8.1 Error counts for DFE(6,1) on Channel A, using SRK algorithm. 200

Table 8.2 Error counts for DFE(6,2) on Channel B, using SRK algorithm. 201

Table 8.3 Error counts for 16 and 64 QAM with variations in ω and noise sequences. 203

Table 8.4 Error counts for DFE(6,g)-II_2, and DFE(6,g)-III with 10% PTS scheme. 209

Table 8.5 Error counts for DFE(6,g)-II_2, and DFE(6,g)-III with RTS scheme. 213

Chapter 1
INTRODUCTION

The high-frequency (HF) radio portion of the electromagnetic spectrum is broadly defined as that range of frequencies running from 2 to 30 MHz. When a beam of electromagnetic radiation at a frequency within this range, originating from a point on the ground, strikes the Earth's ionosphere, it undergoes a process of refractive bending from the ionized layers at different elevations, the net result of which causes a substantial portion of the original beam to be "reflected" back to the ground again. This phenomenon has been exploited for the realization of beyond-the-horizon communication, an information signal of finite bandwidth being modulated onto an HF carrier wave. Ranges up to 2500 miles and beyond can be achieved.

From its inception up to the present time, the greatest interest in HF radio has been for long-distance speech communication as required in military and civilian applications. Analog speech (or voiceband) communication requires a nominal bandwidth of around 3 kHz, the speech signal being modulated in its analog form directly on to the carrier wave. Generally, though, transmission over HF links was considered unreliable because of the fading effects introduced by such channels [1], as will be discussed shortly. The conversion from analog to digital transmission, however, and the use of powerful adaptive signal processing techniques, has offered hope of acceptable network communication quality. In particular, the advent of very-large-scale-integration (VLSI) has brought down the cost of implementation of many of the signal processing techniques that can be used.

The long history of analog speech communication in HF radio has naturally led to this application getting a firm footing in terms of equipment investment and agreements on band allocation. There is motivation, therefore, to considering the transmission of digital traffic over HF channels, whether it be for speech or just information in general, under condition that the bandwidth is ~3 kHz, i.e. the voiceband. Of course larger bandwidths may be used, as in "spread spectrum" receivers, offering improved reception and facilities for information security. In this thesis, however, we shall restrict our attention to voiceband bandwidths.

The increasing availability of satellite communication systems has not led to a demise in the use of HF; in particular, digital HF links offer economic and security advantages over satellite links [1]. Many countries will therefore continue to use HF radio extensively for point-to-point information transfer, commercial shipping, aircraft communication, military land-sea-air operations, etc. [2]. Whereas the main constraint for satellite links is maintaining an adequate signal-to-noise ratio, for HF links it is the dispersion and time-varying characteristics of the transmission medium, which we shall now describe.

1.1 Characteristics of the HF Channel

In most cases the transmitted HF radio waves reach the receiver via more than one distinct path, with the time taken to traverse each path being different. The result is that the impulse response of the channel exhibits a discrete *multipath* structure, with the receiver input, as we shall shortly see, consisting of the linear superposition of phase-shifted-and-attenuated delayed versions of the transmitter output, in addition to channel noise.

Each distinct path from the transmitter to the receiver, by way of the ionosphere, is often termed a *skywave*. The existence of multiple paths is due to [2]

(*i*) skywave returns from different ionospheric layers, namely the E layer (90–110 km high) and the F layer (150–300 km high).

(*ii*) multiple-hops, e.g. skywave may be reflected off the ground and then off the ionosphere again before reaching the receiver.

(*iii*) high and low angle skywave paths, particularly via the F layer.

(*iv*) splitting of the magneto-ionic components, ordinary and extraordinary, resulting from the effects of the Earth's magnetic field.

Although there are several possible routes to the receiver, most of them experience quite a large attenuation. The number of "effective" paths is therefore generally small.

Communication via a line-of-sight path, or *groundwave*, from the transmitter to the receiver is much more straightforward than skywave. The groundwave is just an attenuated and delayed, but otherwise undistorted, version of the transmitted signal. On the other hand a skywave has an attenuation and delay time that can vary with time. In this thesis it will be assumed that all paths to the receiver are from skywaves.

Contrast HF radio with another important transmission medium, troposcatter radio (0.3–10 GHz), which we shall briefly mention. Transmission is brought about by random fluctuations in the dielectric constant of the troposphere [1]. This has the effect of diverting a small fraction of the energy of an incident radio wave back to the receiver. Scatter returns from different points in a "common volume", defined by the transmitter and receiver antennas, have different path delays. Signals scattered from points separated by more than the decorrelation distance of the fluctuations in the dielectric constant are not correlated [1]. The impulse response of the troposcatter channel therefore has a multipath structure, but, unlike the HF channel, the structure is not made up of discrete paths but a continuum of paths. Troposcatter systems will not be considered in this thesis.

Consider a single skywave path which has a delay time of $\tau(t)$ secs. The t variable in $\tau(t)$ expresses the dependence of the quantity on time. In reality a skywave is composed of a number of rays, or sub-paths, that traverse slightly different trajectories [2]. The difference in delay time between the sub-paths is considered to be very much smaller than the reciprocal of the signal bandwidth, so that the receiver is unable to resolve them. Let $\tau_i'(t)$ be the time-varying delay time of the i^{th} sub-path with respect to $\tau(t)$, and also let $\alpha_i(t)$ be the corresponding time-varying attenuation factor for this sub-path. The transmitted signal is a modulated carrier wave of frequency f_c, and can be expressed in general as

$$m(t) = \mathrm{Re}\{x(t)e^{j2\pi f_c t}\} \tag{1.1}$$

where $x(t)$ is the information-bearing part of the signal, assumed to have a bandwidth $B_w \ll f_c$. Ignoring additive noise, the input to the receiver from the skywave is

$$\begin{aligned} c(t) &= \sum_i \alpha_i(t) m(t-\tau_i'(t)-\tau(t)) \\ &= \mathrm{Re}\{\ [\sum_i \alpha_i(t) e^{-j2\pi f_c \tau_i'(t)} x(t-\tau_i'(t)-\tau(t))]\ e^{j2\pi f_c(t-\tau(t))}\} \end{aligned} \tag{1.2}$$

As far as the receiver is concerned, the delayed versions of $x(t)$ from each sub-path cannot be differentiated from one another. Therefore we can assume $x(t-\tau_i'(t)-\tau(t)) \approx x(t-\tau(t))$, giving

$$c(t) \approx \mathrm{Re}\{G(t)x(t-\tau(t))e^{j2\pi f_c(t-\tau(t))}\} \tag{1.3}$$

where

$$\left.\begin{aligned} G(t) &= \sum_i \alpha_i(t) e^{-j\theta_i(t)} \\ \theta_i(t) &= 2\pi f_c \tau_i'(t) \end{aligned}\right\} \tag{1.4}$$

The quantity $G(t)$ is a complex attenuation factor introduced by the skywave; its magnitude is the real attenuation and its argument is a phase shift. The variation in the magnitude of $G(t)$ with time is usually termed *fading*, and it is governed by the time variations of $\{\alpha_i(t)\}$ and $\{\theta_i(t)\}$. The values of $\{\theta_i(t)\}$ can at times result in the real positive attenuations $\{\alpha_i(t)\}$ being added destructively, the value of $|G(t)|$ then being very small or practically zero; at other times they can be added constructively, so that $|G(t)|$ is large. The phase $\theta_i(t)$ will change by 2π rad whenever $\tau_i'(t)$ changes by $1/f_c$. Since $1/f_c$ is a very small number, then only small changes in the ionosphere, such as variations in the height and refractive index of the ionospheric layers [2], are required to produce significant changes in $\theta_i(t)$. In contrast, relatively large dynamic changes in the ionosphere are required for $\alpha_i(t)$ to change sufficiently to cause a significant change in $G(t)$ [3]. Thus the fading rate of $|G(t)|$ is governed primarily by the time variations of $\{\theta_i(t)\}$.

The delays $\{\tau_i'(t)\}$ associated with different sub-paths change at different rates and in a random manner, which implies that $G(t)$ can be modelled as a random process [3]. Studies have shown that the *gaussian-scatter* model for $G(t)$ [4] (to be described in chapter 4) fits experimental data rather well. As well as physical length, the delay time of a skywave path is also dependent on frequency and time [2]. However, if the signal bandwidth is restricted to say 10 kHz or less, and if we consider sufficiently short times of say 10 minutes, most paths are fairly stationary with regard to delay time. Thus HF multipath channels can be adequately represented by models in which the delay time of a path is assumed fixed [2], as it is for example in the gaussian-scatter model. Suppose that there are P paths present, and let $G_i(t)$ and τ_i be, respectively, the complex attenuation factor and delay time of the i^{th} path with respect to the 0^{th} or earliest path. Neglecting the delay of the earliest path , the received signal is

$$c(t) = \text{Re}\{ [\sum_{i=0}^{P-1} G_i(t)x(t-\tau_i)e^{-j2\pi f_c\tau_i}]\, e^{j2\pi f_c t}\} \tag{1.5}$$

It is apparent from (1.5) that the equivalent low-pass received information signal is the expression contained in the square [.] brackets.

Before discussing the design of receiver devices for a fading multipath channel, we will introduce some useful terms that define the characteristics of the channel [1], [3]. Let the impulse response of the channel be described by the two-dimensional random process $h_o(\tau;t)$, with corresponding fourier transform

$$H(f;t) = \int_{-\infty}^{\infty} h_o(\tau;t)e^{-j2\pi f\tau}\, d\tau \tag{1.6}$$

We assume that $h_o(\tau;t)$ is wide-sense-stationary. Then we define the autocorrelation function of $h_o(\tau;t)$ as [3]

$$\Phi_h(\tau_1,\tau_2;\Delta t) = \tfrac{1}{2}\text{E}[h_o^*(\tau_1;t)h_o(\tau_2;t+\Delta t)] \tag{1.7}$$

where the expectation E[.] is over the fading statistics of the channel. It is usually the case that the statistical variations of the paths are independent of one another; this is referred to as *uncorrelated scattering.* Thus we assume that

$$\Phi_h(\tau_1,\tau_2;\Delta t) = \Phi_h(\tau_1;\Delta t)\delta(\tau_1-\tau_2) \tag{1.8}$$

With $\Delta t=0$ the autocorrelation function $\Phi_h(\tau;0)\equiv\Phi_h(\tau)$ is just the average power output of the channel as a function of the delay τ, and it is thus commonly called the multipath intensity profile or delay power spectrum of the channel. The range of values of τ over which $\Phi_h(\tau)$ is essentially non-zero is called the *multipath spread* of the channel; it is a measure of the time between the arrival of the first and last path. We will denote thc multipath spread of the channel as T_m.

Since $h_o(\tau;t)$ is a random process in the t variable, it follows that $H(f;t)$ also has the same statistics. Under the assumption of wide-sense-stationarity, we define the autocorrelation function

$$\Phi_H(f_1,f_2;\Delta t) = \tfrac{1}{2}\mathrm{E}[H^*(f_1;t)H(f_2;t+\Delta t)] \tag{1.9}$$

Using (1.7) and (1.8) it is straightforward to show that

$$\Phi_H(f_1,f_2;\Delta t) = \int_{-\infty}^{\infty} \Phi_h(\tau_1;\Delta t)e^{-j2\pi\Delta f\tau_1}d\tau_1 \equiv \Phi_H(\Delta f;\Delta t) \tag{1.10}$$

where $\Delta f=f_2-f_1$. Thus $\Phi_H(f;\Delta t)$ is the fourier transform of $\Phi_h(\tau;\Delta t)$. Note that it is the assumption of uncorrelated scattering that leads to the autocorrelation function of $H(f;t)$ being dependent only on the frequency difference $\Delta f=f_2-f_1$. For $\Delta t=0$, $\Phi_H(\Delta f;0)\equiv\Phi_H(\Delta f)$ and $\Phi_h(\tau;0)\equiv\Phi_h(\tau)$, with the relationship

$$\Phi_H(\Delta f) = \int_{-\infty}^{\infty} \Phi_h(\tau)e^{-j2\pi\Delta f\tau}d\tau \tag{1.11}$$

The range of values of Δf over which $\Phi_H(\Delta f)$ is essentially non-zero is called the *coherence bandwidth* of the channel; it is a measure of the frequency difference required between two sinusoids for them to be affected differently by the channel [3]. We will denote the coherence bandwidth of the channel as F_c. Because of relationship (1.11) we can assume

$$F_c \approx \frac{1}{T_m} \tag{1.12}$$

If the bandwidth of the transmitted signal, B_w, is such that $B_w \gtrsim F_c$, the channel is said to be *frequency-selective* [1], [3] ($\gtrsim$ means "on the order of or greater than"). In this case the signal is severely distorted by the channel, but there is potential for *implicit frequency diversity gain*, because different parts of the frequency band fade independently [1]. Thus, while one section of the band may be in a deep fade, the remainder can be used for reliable communication. On the other hand, when $B_w \ll F_c$ the channel is said to be *frequency-nonselective*, and the entire frequency band fades with no possible implicit diversity gain. A channel that is frequency-nonselective is unable to resolve the multiple paths, and so to the receiver the transmission appears to have come from only one path.

Consider now the fourier transform

$$\Psi_H(\Delta f;\lambda) = \int_{-\infty}^{\infty} \Phi_H(\Delta f;\Delta t)e^{-j2\pi\lambda\Delta t}d\Delta t \tag{1.13}$$

For $\Delta f=0$, $\Psi_H(0;\lambda)\equiv\Psi_H(\lambda)$ and $\Phi_H(0;\Delta t)\equiv\Phi_H(\Delta t)$, with the relationship

$$\Psi_H(\lambda) = \int_{-\infty}^{\infty} \Phi_H(\Delta t)e^{-j2\pi\lambda\Delta t}d\Delta t \tag{1.14}$$

The quantity $\Psi_H(\lambda)$ is a power spectrum that gives the signal intensity as a function of the *Doppler frequency* λ. The range of values of λ over which $\Psi_H(\lambda)$ is essentially non-zero is called the *Doppler spread* of the channel; it is a measure of the width of the received spectrum when a single sinusoid is transmitted through the channel, or put another way, it is a measure of the rate of change with time of the envelope of the sinusoid. It can equally well be interpreted as the rate of fading of the channel with time. We will denote the Doppler spread of the channel as F_d. The range of values over which $\Phi_H(\Delta t)$ is essentially non-zero is called the *coherence time* of the channel; it is a measure of the minimum difference in time required between the transmission of two sinusoids, of the same frequency, for their respective attenuations suffered to be uncorrelated. We will denote the coherence time of the channel as T_c. The relationship of (1.14) means we can assume

$$F_d \approx \frac{1}{T_c} \tag{1.15}$$

Clearly, for a slow-changing channel, F_d is small, or equivalently, T_c is large.

For a given HF link the maximum usable frequency (m.u.f.) is that frequency above which transmission via the ionosphere is not possible. The m.u.f. is different for different links, tending to be smaller for the shorter range ones. For example, in [2] a particular 430 km link is shown to have an m.u.f.~3 MHz, whereas a longer 1365 km link is shown to have an m.u.f.~12 MHz. Also, as the frequency is reduced below the m.u.f. the number of paths, and consequently the multipath spread, increases to a maximum. This situation tends to be more pronounced with shorter links, where the number of effective paths to the receiver is usually greater. For example, a 2500 km link has been shown [5] to have a maximum multipath spread of about 3 ms; for 1000 km it increases to 5 ms, and for 200 km it is about 8 ms.

1.2 Receiver Designs

Consider a serial digital HF transmission system in which the signalling rate (or baud rate) is T^{-1} Hz. The information signal $x(t)$ transmitted in a particular baud interval takes on any one of a finite number of distinct forms, according to the value of a discrete data symbol variable which represents a finite number of bits of information. Let the number of bits transmitted per baud interval be $\log_2 M$, in which case the number of discrete values and forms of the data symbol variable and $x(t)$ respectively is M. Also, let B_s denote the available bandwidth (in Hz) at our disposal, which is assumed to be an integer multiple (≥ 1) of the bandwidth B_w of $x(t)$. In determining the signalling strategy, by which the information signal $x(t)$ in a particular baud interval takes on any one of M distinct forms, it is important to consider the efficiency index E_f given by

$$E_f = \frac{R}{B_s} \frac{(\text{bits s}^{-1})}{(\text{Hz})} \tag{1.16}$$

where R is the data rate, in bits s^{-1}, transmitted through the channel of bandwidth B_s Hz. When the data rate is such that $E_f < 1$, the appropriate strategy is to select each of the M waveforms to be orthogonal to one another [3], i.e.

$$\int_0^T x_i(t)x_k^*(t)dt = 0 \qquad \text{for } k\neq i$$
$$\neq 0 \qquad \text{for } k=i \tag{1.17}$$

where $x_i(t)$, $1\leq i\leq M$, identifies a particular waveform out of the set of M. An example of an orthogonal signalling scheme is orthogonal frequency-shift-keying (FSK) [3]. Receiver designs that are based on the condition $E_f<1$ are usually termed *low-data rate receivers.* When $E_f\gtrsim 1$, the appropriate strategy is one which is bandwidth-efficient; that is, for a fixed bandwidth the rate R increases logarithmically with an increase in the number of waveforms M [3]. Linear modulation schemes like single-sideband-pulse-amplitude-modulation (SSB-PAM), quadrature-amplitude-modulation (QAM), and phase-shift-keying (PSK) are bandwidth-efficient, and therefore suitable for high speed data transmission. Receiver designs that are based on the condition $E_f\gtrsim 1$ are usually termed *high-data rate receivers.*

Digital speech requires a data rate of about 2.4 kbits s^{-1} for acceptable quality [2]; with a bandwidth of 3 kHz this would imply an E_f of 0.8. Thus most HF voiceband systems employ a bandwidth-efficient linear modulation scheme. In this thesis we will be concerned with data rates much greater than 3 kbits s^{-1}; it can be assumed, therefore, that a linear modulation scheme is employed, so that the information signal in any baud interval consists of a fixed pulse of bandwidth B_w multiplied by a data symbol that takes on any one of M discrete values [3]. For QAM and PSK the fixed pulse is real, while for SSB-PAM it is one sideband of a real pulse, therefore making it complex. For SSB-PAM the data symbol is real-valued, while for QAM and PSK it is in general complex-valued. Unless stated otherwise, it should be assumed that the discussion is with reference to QAM and PSK.

We assume that at the transmitter the fixed real pulse in any baud interval does not significantly interfere with the pulses in other neighbouring baud intervals, i.e. there is no significant *intersymbol interference* (ISI). It is therefore required that

$$B_w \gtrsim \frac{1}{T} \tag{1.18}$$

In most voiceband applications bandwidth is a precious commodity, so it is usual to have

$$B_w \sim \frac{1}{T} \tag{1.19}$$

i.e transmit as close to the Nyquist rate as possible without incurring significant ISI. Unless stated otherwise, condition (1.19) will be assumed to hold. The data rate R is given by

$$R = \frac{B_s}{B_w}.\frac{\log_2 M}{T} \sim B_s \log_2 M \tag{1.20}$$

where it is assumed that the bandwidth B_s can be sub-divided into a number of sub-channels of bandwidth B_w, each independently transmitting $\log_2 M$ bits per baud interval T. Note that for SSB-PAM there needs to be an extra multiplying factor of 2 on the LHS of (1.18) and (1.19), and therefore also on the RHS of (1.20), because B_w in this case is only one half of the bandwidth of a real pulse.

We will now discuss the design of receivers firstly under condition that there be no ISI at the receiver, and secondly with this condition relaxed.

1.2.1 Receivers for Channels with Negligible ISI

Given a fixed amount of received signal energy, for a time-invariant channel it is desirable, both in terms of performance and simplicity of receiver design, for the received signal to be just a delayed and attenuated, but otherwise undistorted, version of the transmitter output. In the case being studied here, this situation can be approximated to by making the baud interval T significantly greater than the multipath spread of the channel, i.e.

$$T \gg T_m \tag{1.21}$$

$$\Rightarrow \quad B_w \ll F_c \tag{1.22}$$

Thus each sub-channel is frequency-nonselective, and therefore the receiver effectively sees only one skywave path in each sub-channel. As well as by making T large enough, condition (1.21) can be realized by judicious choice of the operating frequency with respect to the m.u.f., so as to limit the amount of multipath spread. Another insurance against significant ISI is to employ time-guard bands at the end of each baud interval, the duration of which should be $\sim T_m$.

Consider the data rate of 2.4 kbits s^{-1} required for digital voice. Substituting B_s=3 kHz in (1.20) we get $R\sim3000.\log_2 M$; thus M=2 would provide us with acceptable voice quality. Typically, $T_m\sim1$ ms for HF, and so T should be at least about 10 ms to satisfy (1.21). This makes B_w at most about 100 Hz, and therefore calls for a type of parallel sub-channel transmission system to be used. One of the earliest such systems was Kineplex [2], [6]. One version of this uses 20 sub-channels, spaced 100 Hz apart, each one carrying a serial 75 baud s^{-1} transmission of four-phase signals (equivalent to 2 bits per baud interval); the total band occupancy is therefore 2 kHz and the data rate is 3 kbits s^{-1}. The system employs differential-phase-shift-keying (DPSK), or to be more precise, the actual information is coded/decoded as the phase difference between baud-adjacent four-phase signals. For correct operation, this strategy of DPSK requires that the channel state be very slowly changing with respect to the signalling rate, i.e.

$$\frac{1}{T} \gg F_d \tag{1.23}$$

therefore ensuring that the phase shift introduced by the time-varying medium is virtually the same for signals in adjacent baud intervals. Fortunately, F_d is usually no more than just a few Hz on most HF channels, and so (1.23) can be assumed to hold in general for all the receivers we discuss. When condition (1.23) holds, the channel is described as being slow-fading. The Kineplex is classed as a non-adaptive system, since at no time is compensation made for the time-varying effects of the channel.

Another non-adaptive parallel sub-channel system is Andeft [2], [7], the essential difference with Kineplex being that the DPSK coding scheme is between simultaneous (in time) signals in frequency-adjacent sub-channels. For correct operation, this strategy of DPSK therefore requires that

$$B_w \ll F_c \tag{1.24}$$

so that signals in adjacent sub-channels do not experience significantly different phase shifts. The Andeft uses 66 sub-channels at a frequency spacing of 40 Hz, which compares favourably with F_c which is typically ~1 kHz. Remembering (1.19), condition (1.24) is actually no more than what we assumed to begin with in (1.21).

It can be seen that DPSK is inferior to coherent (i.e. phase compensated) detection of PSK, since just one error in a string of coherently detected PSK signals gives rise to two errors if the differential-phase scheme is adopted. In [3] it is shown that for a frequency-nonselective Rayleigh fading channel, the advantage of binary coherent PSK signalling is 3 dB, in terms of signal-to-noise ratio or probability of error, over the corresponding DPSK signalling.

However, as is shown in [3], perhaps the most important feature of transmission on frequency-nonselective fading channels is that the error rate decreases only inversely with the signal-to-noise ratio, in contrast to an exponential decrease that occurs with time-invariant channels. This means that the transmitter must transmit a large amount of power in order to obtain a low probability of error, which in many cases is not possible either technically or economically. This leads us on to a discussion of diversity techniques as a means to improving performance.

System performance is influenced primarily by the deep fades that occur during transmission. Diversity techniques are basically schemes in which redundancy is introduced into the system in an attempt to lessen the chance of severe signal loss. This takes the form of providing the receiver with multiple replicas of the same information signal, each of which have arrived under independently fading channel conditions. Thus, for example, if p is the probability that any one signal fades below some critical value, then p^N is the probability that N independently fading replicas of the same signal will fade below the critical value.

An obvious method that comes to mind for achieving diversity is to employ multiple receiving antennas. The receiving antennas have to be spaced sufficiently far apart for their respective received signals to fade independently of one another; a separation of at least 10 wavelengths [3] is usually required between any two antennas. This method, however, seems a bit physically cumbersome, and costly too, for HF channels, since a frequency of 30 MHz would require an antenna spacing ~100 m.

The multiple-antenna method just described is an example of what is sometimes referred to as *explicit diversity* [1], because of the externally visible nature of the technique. The alternative form of diversity, i.e. *implicit diversity*, is when the channel itself provides the redundancy. For example, if the channel is frequency-selective, i.e.

$$B_s \gtrsim F_c \tag{1.25}$$

then there is potential for implicit frequency diversity gain, because different portions of the frequency band fade independently of one another. Typically, $F_c \sim 1$ kHz, implying that voiceband channels have potential for implicit frequency diversity gain. For parallel sub-channel systems, frequency diversity can be realized by transmitting the same information on different sub-channels that are sufficiently far apart in frequency. This is exactly what is done in the Kathryn system [8]. Unfortunately, introducing diversity in this fashion does decrease the overall net information rate of the system, since sub-channels that could otherwise be carrying useful information are instead carrying redundant information.

Another type of diversity is *implicit time diversity*, which is related to the coherence time T_c. In fast-fading channel environments, where $T^{-1} \gtrsim F_d$, redundant symbols in a coding scheme can be used to provide time diversity if the code word spans more than one fade period. In slow-fading environments, as characterized by (1.23), this condition of spanning the fade period can be realized by interleaving the code words so as to introduce time gaps $\gtrsim T_c$ between successive symbols in a particular code word. However, apart from reducing the net information rate of the system, this scheme means that the signal delay will be greater than T_c. In practice, for such applications as transmission of digitized speech, the required time delay is unsatisfactorily long for two-way communication [1]. Thus the interest in providing implicit diversity tends to be focussed mainly in the frequency domain.

Before leaving this section, we will digress a little to mention an important type of wideband receiver that utilizes frequency diversity through the availability of a very large bandwidth, such that

$$B_w \gg F_c \gg \frac{1}{T} \tag{1.26}$$

This means that the signal pulses are so narrow that the receiver can resolve all the path components of the received signal. Thus, within each baud interval, there are P time-displaced versions of the same information signal. A device commonly employed for the optimal combining of the information signal from all the independently fading replicas is a *RAKE receiver* [3], [9]. RAKE receivers can be made non-coherent through the use of DPSK signals, although versions using coherent PSK signals are still superior in performance [3], assuming the phase compensation is perfect. The frequency diversity present in wideband systems gives rise to a significant improvement in performance over frequency-nonselective channels. However, since voiceband bandwidths do not really allow us to realize (1.26), wideband receivers like the RAKE will not be considered in this thesis.

1.2.2 Receivers for Channels with ISI

The parallel sub-channel systems discussed in the last section were originally adopted for voiceband HF because of the desire to have $T \gg T_m$. Suppose now that we relax this condition, and consider the serial transmission of data over the whole of the voiceband, so that

$$\frac{1}{T} \sim B_w = B_s \gtrsim F_c \tag{1.27}$$

$$\Rightarrow \qquad T \lesssim T_m \tag{1.28}$$

Therefore there is intersymbol interference at the receiver. Receiver devices that are designed to mitigate the effect of ISI are broadly termed *equalizers.* A specific requirement of these devices is that they must know the channel impulse response, or some quantity related to it, and so in a time-varying environment it is essential they be made adaptive. An important advantage of digital systems is that the transmitted source symbols only have a finite number of values. Given a high likelihood that receiver decisions on these symbols are correct, they can therefore be used as a form of sounding signal to follow the channel's variations, and thereby provide what is referred to as *decision-directed adaptation* [1]. Thus no power need be used on the transmission of special pilot or probe signals for tracking the channel. The adaptation process does, however, possess some inertia, and so performance of the equalizer is critically dependent on the degree to which condition (1.23) holds; the slower the slow-fading channel is, the better the adaptation. Equalizers, and the algorithms used to make them adaptive, will be described in detail in later chapters.

It will be noted from (1.27) that there is some degree of resolution of the paths at the receiver, or put another way, there is implicit frequency diversity. The important feature of adaptive equalizers, under such circumstances, is that when correctly adjusted they can cope with ISI and still wind up with a net diversity gain [1]. Also, the diversity gain is achieved without having to deliberately reduce the net information rate, as is done for parallel sub-channel systems, essentially because the constraint of (1.21) no longer applies. It has been shown that serial transmission systems with implicit frequency diversity, assuming the adaptation of the equalizers is perfect, can offer substantially better performance than parallel sub-channel systems [10], [11]. Other disadvantages of parallel sub-channel systems, relative to serial systems, are the possibility of co-channel interference, and [12]

(*i*) inefficient use of available transmitter power. The amplitude of the transmitted signal varies according to the relative phases of the signals in the sub-channels. The random nature of the signals in each sub-channel causes the composite transmitted signal to have infrequent peaks of magnitude several times the mean level. Since transmitter amplifiers are peak-power limited, the average power output is likely to be much lower than the maximum available. In [13] it is reported that for a 16 sub-channel system, the average power output was 100 W from a transmitter capable of producing 1 kW.

(*ii*) the need for time-guard bands, which reduces the information capacity of the system. For example, the Kathryn system [8] employs a 1 ms time-guard in a baud interval of 13.3 ms; thus the information capacity is reduced by about 8%. Alternatively, for a given baud interval, the presence of guard bands reduces the number of sub-channels possible within an allocated bandwidth.

1.3 Objectives and Outline of the Thesis

This thesis is concerned with the serial transmission of data over a voiceband HF channel. Before being more specific on this, we will discuss some notable papers published prior to, and during the early part of, this study. These are related to the work undertaken in this thesis.

1.3.1 Related Work

In [14] the error rate performance of bandwidth-efficient modulation techniques, like 8-ary PSK and 8-ary PAM-PSK, are examined for a serial data 3 kbaud s^{-1} HF channel, both two-path and three-path. The receiver uses a decision feedback equalizer (DFE). The results show that at data rates as high as 9 kbits s^{-1}, and with perfect adaptation of the DFE, energy-efficient constellations involving a combination of PAM and PSK yield better performance than straightforward PSK.

In [15] the performance of an adaptive DFE is observed for a two-path 3 kbaud s^{-1} HF channel. The parameters of the DFE are directly adjusted by a tracking algorithm, this method being the conventional approach to implementing an adaptive DFE. Two types of decision-directed tracking algorithm are compared; the steepest descent (SD) algorithm and the Kalman algorithm. With correct decisions being fed to the tracking algorithms, the results are strongly in favour of the Kalman algorithm; the SD algorithm produces an irreducible error rate that is many orders of magnitude greater than the error rate for the Kalman algorithm. When actual receiver decisions are fed to the Kalman tracking algorithm, it is shown that a request-for-training-sequence (RTS) scheme is useful in maintaining performance, particularly when tracking suffers, as happens in a deep fade or when the channel changes abruptly. The largest number of signal points (i.e. data symbol levels) used in [15] is 8, e.g. 8-ary PSK.

Reference [16] contains the derivation for a square-root formulation of the recursive least squares (RLS) tracking algorithm, the RLS algorithm being just a variant of the Kalman algorithm. It also derives a modified version of the algorithm to give improved performance. More will be said concerning ref. [16] later in the thesis. Performance results of the transmission of 8-ary PSK signals over a 2.4 kbaud s^{-1} two-path HF channel re-affirm results given in [15]; essentially, that with correct decisions being fed to the adaptive algorithm, the signal-to-noise ratio loss with respect to the situation in which perfect adaptation is assumed is reasonably small, e.g. 2.0 dB at an error rate of 10^{-4} with a 1 Hz fade rate.

Whilst [14]–[16] have investigated the performance of the DFE, [11], [17] and [18] have investigated the performance of another equalizer device, namely the maximum likelihood sequence estimator (MLSE). Actually, because of the complexity of the MLSE, an approximation to this device is often used, when it is then referred to as a near-MLSE. The observed performances of the near-MLSE's in [17] and [18] are primarily under condition that the adaptation is perfect. The argument for adopting the MLSE/near-MLSE lies essentially in the fact that, under condition of perfect knowledge of the channel state, the MLSE invariably outperforms the DFE, although the degree to which it does so is highly dependent on the channel impulse response. The argument for the DFE is its much lower cost of implementation. Very little has appeared in the literature in which the two devices are compared under the same fading channel conditions. In this thesis we will remedy this to some extent, and in doing so give some idea of the magnitude of the trade-offs between the DFE and MLSE/near-MLSE.

In [11] and [17] the data rate is set at 2.4 kbits s^{-1}, the signalling rate being 1.2 kbaud s^{-1} with 4 QAM (equivalent to 4-ary PSK) signals, over a two-path channel. In [18], 16 QAM signals are transmitted at 2.4 kbaud s^{-1} over a two-path channel, thus giving a data rate of 9.6 kbits s^{-1}. In this thesis we shall observe performance with 64 QAM signals at 2.4 kbaud s^{-1}, giving a data rate of 14.4 kbits s^{-1}; this data rate is much higher than anything previously looked at for voiceband HF channels.

In all the references mentioned in this section, the HF transmission system is simulated in the laboratory using a statistical model.

1.3.2 Objectives of the Thesis

The primary objective of the research in this thesis is to investigate the transmission of 4, 16 and 64 QAM data signals at 2.4 kbaud s^{-1} over a voiceband HF radio link; the data rate respectively being 4.8, 9.6 and 14.4 kbits s^{-1}. The signalling rate is such that ISI is present at the receiver, thus requiring the employment of, and therefore the focussing of attention on, adaptive equalizers. Two ISI-mitigating devices are examined; namely, the DFE and the MLSE/near-MLSE. Their performance is compared using a computer simulated model of a two-path and three-path HF channel, both of which have additive gaussian noise. The performances offered by two decision-directed tracking algorithms are looked at, these being the SD and RLS.

The performances of the equalizer devices are observed under three conditions, or *modes*. The first (mode I) is where the adaptation of the equalizer is perfect; the second (mode II) is where the data symbol values employed in the tracking algorithm are equal to the values of the transmitted data symbols; the third (mode III) is where the tracking algorithm uses actual receiver decisions on the transmitted data symbols. The three modes are explained in more detail at the beginning of chapter 5. By examining the performance in each mode and comparing, we can ascertain the degradation caused by the use of a tracking algorithm and also the effect of incorrect receiver decisions. This will help to pinpoint the respective tolerances of each receiver device to adaptation error, and give an appreciation of what steps need to be taken in order to produce acceptable performance, especially at the higher data rates. In mode III the usefulness of having training sequences from the transmitter, as a means to maintaining and even improving performance, is examined. In some cases we have employed an analytical bound for measuring the error rate of the DFE; this is useful, as it allows us to measure error rates much lower than can be recorded by the commonly used "error count" approach. Further details will be given in chapter 5.

The investigations have produced some interesting by-products. A new way of implementing the adaptive DFE, and also the pre-filter of a near-MLSE, has been developed. The method is based on estimating (i.e. tracking) the channel sampled impulse response (*channel estimation*), and is shown to have advantages over the conventional way of implementing the DFE. The process of channel estimation has been mathematically analysed for both the SD and RLS algorithms, and the accuracy of the algorithms is compared for both theory and experiment. A method is shown by which the performance of the adaptive DFE can be predicted, using the new way of implementing it. Also, the modified version of the RLS algorithm given in [16] is shown to be unstable under certain conditions.

1.3.3 Brief Outline of Chapters

Chapters 2–4 are basically review material and groundwork for the thesis. Chapters 5–8 contain all the results of our investigations.

Chapter 2 is a review of the basic elements of a QAM transmission system, and contains details of various receiver devices used to combat ISI.

Chapter 3 contains a description of the two tracking algorithms used in this thesis, the SD and RLS, and a discussion of their respective merits.

Chapter 4 describes the gaussian-scatter model for an HF channel, and gives details of the actual simulation model, fade rates, time-spans etc. used in the thesis.

Chapter 5 is essentially concerned with performance in mode I. It presents comparisons of the

DFE with the ideal matched filter equalizer, and also the DFE with the MLSE/near-MLSE, revealing some interesting trade-offs between the devices. Chapter 5 also contains a few other interesting details, some of which have already been mentioned.

Chapter 6 investigates the capabilities of the SD and RLS algorithms, when used for the task of channel estimation in the HF systems of this thesis. A mathematical analysis of the performances of the two algorithms is presented, including a derivation for the optimum setting of their respective adaptation parameters. Our theoretical derivations are compared with experiment.

Chapter 7 is concerned with performance in mode II. It describes in detail a method by which the adaptive DFE can be implemented via the channel estimate, and compares this new approach with a conventional RLS approach. The modified form of RLS tracking algorithm given in [16] is analysed and shown to be unstable under certain conditions. Using the new way of implementation, a method is given by which simple predictions of the performance of the DFE can be made. Results comparing the performances of the MLSE/near-MLSE and DFE are shown and discussed.

Chapter 8 looks at the performance of the DFE in mode III. Besides the no-training case, two approaches to the employment of training sequences from the transmitter are also examined. One training approach in particular is shown to provide significant gains.

Chapter 9 contains a summary of the findings of the thesis, with additional comments, and suggestions for further work.

Chapter 2
DETECTION TECHNIQUES IN A QAM TRANSMISSION SYSTEM

In this chapter we will first describe the basic elements of a QAM transmission system. Then, assuming ISI is present at the receiver, we will give detailed derivations of the most commonly used equalizer devices for mitigating ISI, and discuss their relative merits. We begin by assuming a time-invariant channel, for simplicity, and later extend the analysis to time-varying channels.

2.1 The QAM Transmission System

Linear modulation schemes like SSB-PAM (or its more practical version, vestigial-sideband (VSB) PAM [19]) and QAM are well suited for high-speed data transmission because of their efficient use of available bandwidth [19], [20]; to make full use of the bandwidth, however, it is necessary for the receiver to employ a coherent linear demodulation scheme. QAM signals have an advantage over PAM signals in that the knowledge of the carrier phase can be acquired in an easier manner, as we will mention later. QAM signals are as efficient in terms of bits s^{-1} per Hz of bandwidth as SSB-PAM signals (with VSB-PAM signals being slightly less efficient). A QAM signal is actually a combination of PAM and PSK, and is generally more efficient in its use of energy than is a pure PSK signal with the same number of levels.

Fig. 2.1 shows the basic components of a QAM transmission system. At the input are two data streams $\sum_i s_{i1}\delta(t-iT)$ and $\sum_i s_{i2}\delta(t-iT)$, where s_{i1} and s_{i2} can take any of the L integer values $-(L-1)$, $-(L-3)$, ... , $(L-3)$, $(L-1)$ where L is even, and T is the baud interval. The pulse-shaping low-pass filters (LPF's) for each of the data streams are identical, with causal impulse response $a(t)$. The outputs of the two filters are each modulated by a carrier of frequency f_c, with a mutual phase difference of 90°, and summed to give the QAM signal

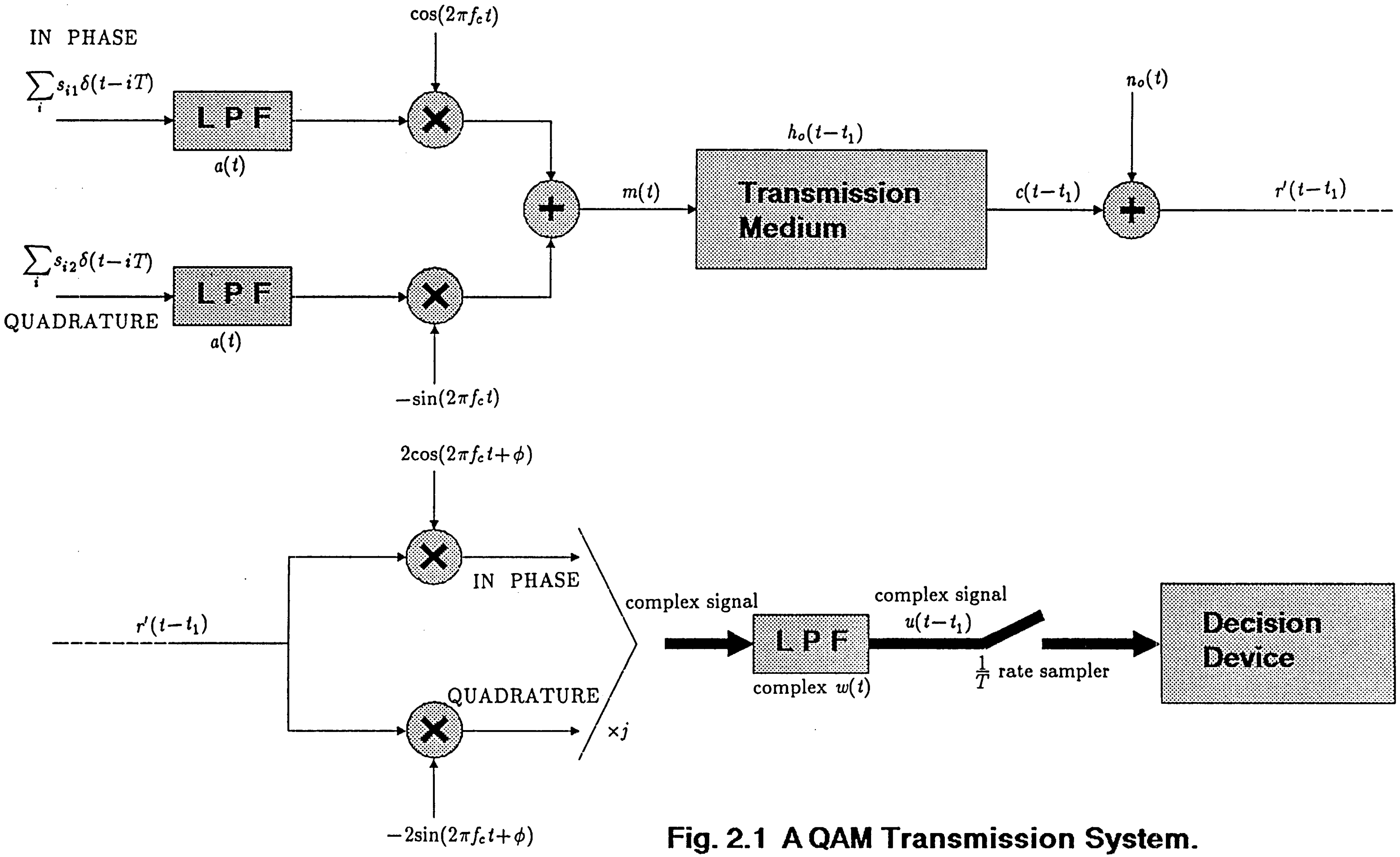


Fig. 2.1 A QAM Transmission System.

$$m(t) = [\sum_i s_{i1} a(t-iT)]\cos 2\pi f_c t - [\sum_i s_{i2} a(t-iT)]\sin 2\pi f_c t$$

$$= \mathrm{Re}\{ [\sum_i s_i a(t-iT)]\, e^{j2\pi f_c t} \} \tag{2.1}$$

where $s_i = s_{i1} + js_{i2}$ (complex). Therefore

$$m(t) = \mathrm{Re}\{x(t) e^{j2\pi f_c t}\} \tag{2.2}$$

where

$$x(t) = \sum_i s_i a(t-iT) \tag{2.3}$$

The fourier transform (FT) of $a(t)$ is $A(f)$, and is such that

$$|A(f)| = 0 \quad \text{for} \quad |f| > f_c \tag{2.4}$$

which implies that $X(f)$, which is the FT of complex baseband data signal $x(t)$, is such that

$$|X(f)| = 0 \quad \text{for} \quad |f| > f_c \tag{2.5}$$

It will be assumed in this thesis that the two data streams are statistically independent, and that each s_{i1} and s_{i2} for any i is equally likely to take on any one of its allowed integer values. The complex data symbol s_i has L^2 possible values, and the system is therefore called an L^2 QAM transmission system. Letting E[.] denote the statistical mean, we have for any i

$$\mathrm{E}[s_{i1}] = \mathrm{E}[s_{i2}] = 0$$

$$\mathrm{E}[s_{i1}^2] = \mathrm{E}[s_{i2}^2] = \tfrac{1}{3}(L^2-1)$$

$$\mathrm{E}[|s_i|^2] = \mathrm{E}[s_{i1}^2] + \mathrm{E}[s_{i2}^2] = \tfrac{2}{3}(L^2-1) \tag{2.6}$$

It can be seen that the data signal $x(t)$ has no d.c. component, which means that the QAM signal $m(t)$ is "suppressed-carrier".

The impulse response of the transmission medium is denoted as $h_o(t-t_1)$, where t_1 is the delay in sending a pulse through the channel ($h_o(t)$ is causal). The signal $c(t-t_1)$ at the output of the medium is given by

$$\begin{aligned} c(t-t_1) &= m(t)\bullet h_o(t-t_1) \qquad\qquad (\bullet \equiv \text{convolution}) \\ &= \mathrm{Re}\{\ [x(t)e^{j2\pi f_c t}]\bullet h_o(t-t_1)\ \} \\ &= \mathrm{Re}\{\ [x(t)\bullet(h_o(t-t_1)e^{-j2\pi f_c t})]\ e^{j2\pi f_c t}\} \\ &= \mathrm{Re}\{\ [x(t)\bullet h(t-t_1)]\ e^{j2\pi f_c t}\} \end{aligned} \tag{2.7}$$

where

$$h(t) = h_o(t)e^{-j2\pi f_c(t+t_1)} \tag{2.8}$$

The signal at the receiver input is

$$r'(t-t_1) = c(t-t_1) + n_o(t) \tag{2.9}$$

where $n_o(t)$ is additive noise. If the in-phase component is represented as the real part of a complex quantity, and the quadrature component as the imaginary part, then the complex signal $u(t-t_1)$ at the output of the complex LPF $w(t)$ in fig. 2.1 is given by

$$\begin{aligned} u(t-t_1) &= \{r'(t-t_1)2e^{-j(2\pi f_c t+\phi)}\}\bullet w(t) \\ &= \Big\{[x(t)\bullet h(t-t_1)]e^{-j\phi}+[x(t)\bullet h(t-t_1)]^{*}e^{-j(4\pi f_c t+\phi)}\Big\}\bullet w(t) \\ &\qquad\qquad +[2n_o(t)e^{-j(2\pi f_c t+\phi)}]\bullet w(t) \end{aligned} \tag{2.10}$$

Assume $w(t)$ rejects the double-frequency $e^{-j4\pi f_c t}$ signal components, giving

$$u(t-t_1) = \Big\{[x(t)\bullet h(t-t_1)]e^{-j\phi}+n(t-t_1)\Big\}\bullet w(t) \tag{2.11}$$

where

$$n(t-t_1) = 2n_o(t)e^{-j(2\pi f_c t+\phi)} \tag{2.12}$$

The quantity ϕ is the phase difference between the transmitter and receiver carriers. The filter $w(t)$ also has another function, that of noise suppression. The complex data symbols $\{s_i\}$ are determined from the periodic T-spaced samples of $u(t-t_1)$, and in the next section optimum characteristics for $w(t)$ will be derived under certain criteria. However, it is obvious that such a $w(t)$ should at least filter out all noise beyond the bandwidth of $x(t)$, and therefore, recalling (2.5), the assumption that $w(t)$ rejects the double-frequency components is not restrictive.

Compare the QAM system of fig. 2.1 with the SSB-PAM system of fig. 2.2. QAM transmits at twice the bit rate but uses twice the bandwidth. It is therefore as efficient in bits s^{-1} per Hz as SSB-PAM. Notice that to avoid a reduction in signal power for SSB-PAM (and also VSB-PAM), the receiver demodulating carrier should be in phase synchronism with the transmitter modulating carrier, requiring the transmission of a pilot carrier to achieve this, at the expense of greater equipment complexity [21]. In contrast, QAM signals do not require the receiver carriers to be in any special phase relationship with the transmitter carriers. Any phase difference can be compensated for using data-aided techniques, as will be described in chapter 4.

Let
$$y(t) = [a(t) \bullet h(t)]e^{-j\phi} \tag{2.13}$$

Then
$$u(t-t_1) = \Big\{\sum_i s_i y(t-t_1-iT)+n(t-t_1)\Big\} \bullet w(t) \tag{2.14}$$

If the transmission medium characteristic $h(t)$ changes with time, as is the case for an HF radio channel, and/or if ϕ has time variations, then it is necessary for $w(t)$ to be made adaptive.

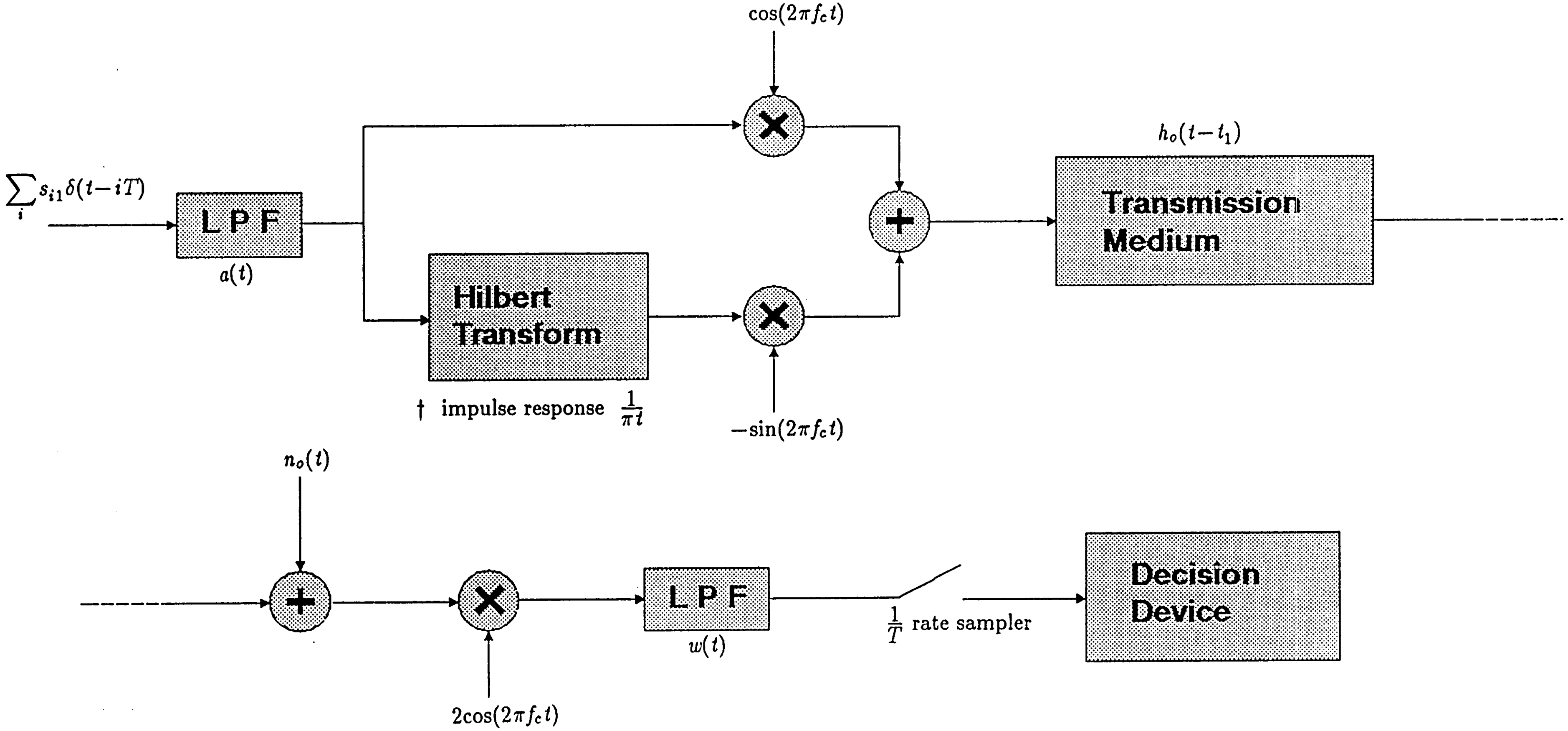
Let
$$b(t) = y(t) \bullet w(t) \tag{2.15}$$

The sample of $u(t-t_1)$ at time t_2+kT, where k is an integer, is

$$\begin{aligned} u(t_o+kT) &= \sum_i s_i b(t_o+(k-i)T)+n_w(t_o+kT) \\ &= s_k b(t_o)+\sum_{i\neq 0} s_{k-i} b(t_o+iT)+n_w(t_o+kT) \end{aligned} \tag{2.16}$$

where
$$t_o = t_2-t_1 \quad \text{and} \quad n_w(t) = n(t) \bullet w(t) \tag{2.17}$$

The quantity t_o is termed the sampling phase. In the absence of noise, in order to detect the $\{s_i\}$ both $b(t)$ and t_o must be known. In the detection of s_k, the middle term (i.e. the



† frequency response of $\frac{1}{\pi t}$ is

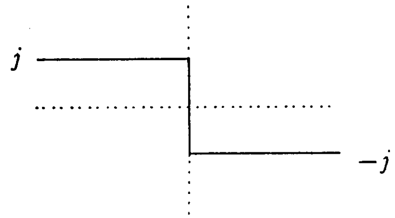


Fig. 2.2 An SSB-PAM Transmission System.

summation) of (2.16) is interference from neighbouring symbols (ISI), while the last term is interference from additive noise. The equivalent baseband QAM model based on (2.14), (2.15) and (2.16) is shown in fig. 2.3. All signals are complex, and $y(t)$ is causal.

The additive noise $n(t)$ in fig. 2.3 shall henceforth be assumed to be a stationary complex white gaussian random process, with zero mean, the real and imaginary parts being independent stationary white gaussian random processes with zero mean and equal variance of $\sigma_n^2/2$. This is a common assumption in the theoretical analysis and simulation of communication systems such as voiceband [3], made not just for the mathematical convenience, but also because it generally gives a fair representation of the real performance to be expected [21].

2.2 Equalization Strategies at the Receiver

Equation (2.16) can be written as

$$u_k = b'_o s_k + \sum_{i \neq 0} s_{k-i} b'_i + n_{wk} \tag{2.18}$$

where

$$\begin{aligned} u_k &= u(t_o + kT) \\ b'_i &= b(t_o + iT) \\ n_{wk} &= n_w(t_o + kT) \end{aligned}$$

Suppose that the decision device is to detect s_k from u_k and that it has at its disposal the values of $\{s_{k-i}\}$ for $-N_1 \leq i \leq -1$, $1 \leq i \leq N_2$. Then the input to the threshold detector, $\tilde{s}_k$, that produces the estimate $\hat{s}_k$, is given by

$$\tilde{s}_k = u_k - \sum_{\substack{i=-N_1 \\ i \neq 0}}^{N_2} s_{k-i} b_i \tag{2.19}$$

The error in $\tilde{s}_k$ is

$$\begin{aligned} e_k &= \tilde{s}_k - s_k \\ &= s_k(b'_0 - 1) + \sum_{\substack{i=-N_1 \\ i \neq 0}}^{N_2} s_{k-i}(b'_i - b_i) + \sum_{i=-\infty}^{-(N_1+1)} s_{k-i} b'_i + \sum_{i=N_2+1}^{\infty} s_{k-i} b'_i + n_{wk} \end{aligned} \tag{2.20}$$

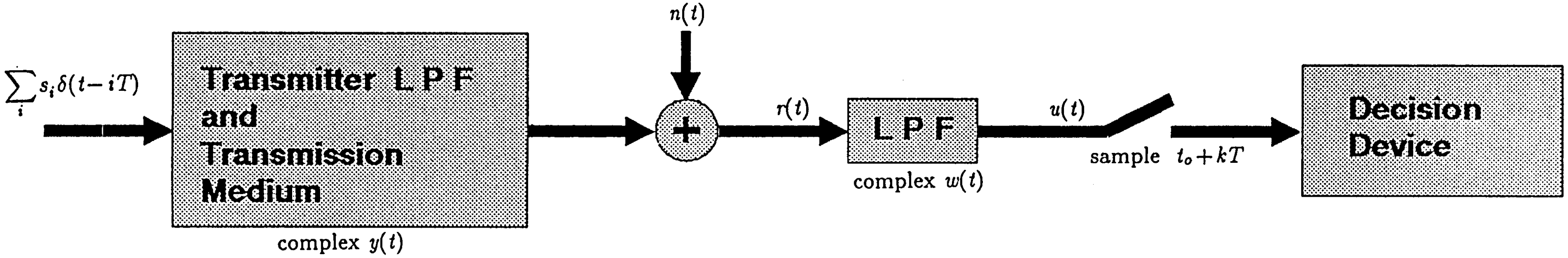


Fig. 2.3 QAM Baseband Transmission System.

The mean square error (MSE), denoted as $\xi(N_1,\ N_2)$, is

$$\begin{aligned}\xi(N_1,\ N_2) &= \mathrm{E}[|e_k|^2] \\ &= \sigma_s^2\Big\{|b_0'-1|^2+\sum_{\substack{i=-N_1\\ i\neq 0}}^{N_2}|b_i'-b_i|^2+\sum_{i=-\infty}^{-(N_1+1)}|b_i'|^2+\sum_{i=N_2+1}^{\infty}|b_i'|^2\Big\}+\sigma_{nw}^2 \end{aligned} \tag{2.21}$$

where

$$\sigma_s^2 = \mathrm{E}[|s_i|^2] = \tfrac{2}{3}(L^2-1)$$

$$\sigma_{nw}^2 = \mathrm{E}[|n_{wk}|^2]$$

Two popular criteria of optimality for the selection of $w(t)$ and $\{b_i\}$, $(-N_1 \leq i \leq N_2,\ i\neq 0)$, are:

(i) the *zero-forcing* (ZF) criterion. This minimizes $\xi(N_1,\ N_2)$ subject to the condition that the interference from data symbols, which is the term in $\{.\}$ on the RHS of (2.21), is zero. This definition assumes that there is no restriction on the realization of $w(t)$. The solution can be expressed in general as the cascade of a band-limited continuous filter and a transversal filter. In practice the transversal filter is implemented as a digital filter, which can only have a finite number of taps, which therefore means that the interference from data symbols cannot in general be zero; thus, in this context, the ZF criterion requires only a certain range of interference terms in (2.21) to be zero, e.g. see [3]. Whenever the ZF criterion is mentioned in this thesis, it should be assumed that it refers to the case of no restriction on $w(t)$, so that all the interference from data symbols is zero. Actually, though it is not assumed here, the term "zero-forcing" is often applied only to the condition of having no interference from data symbols, irrespective of whether ξ is minimized.

(ii) the *minimum MSE* (MMSE) criterion. This simply minimizes $\xi(N_1,\ N_2)$. Obviously MMSE provides a lower ξ than does ZF, though the two solutions will tend to coincide as the noise power gets smaller. This thesis will mainly consider MMSE, as it seems to be the more widely used criterion [3], probably because it generally lends itself more easily to implementation by means of recursive tracking algorithms (to be discussed in chapter 3).

The term "intersymbol interference" is, strictly speaking, supposed to mean the interference from neighbouring symbols, although when we use it we shall also mean it to include the interference from the current symbol being detected, as represented by the $|b'_0-1|^2$ term on the RHS of (2.21).

The noise power σ_{nw}^2 can be written as

$$\sigma_{nw}^2 = \mathrm{E}[|\int_{-\infty}^{\infty} n(t_o+kT-u)w(u)\ du|^2]$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{E}[n(t_o+kT-u)n^*(t_o+kT-x)]w(u)w^*(x)\ du\ dx \qquad (2.22)$$

From the definition of $n(t)$,

$$\mathrm{E}[n(t_o+kT-u)n^*(t_o+kT-x)] = \sigma_n^2\delta(u-x) \qquad (2.23)$$

$$\Rightarrow \qquad \sigma_{nw}^2 = \sigma_n^2\int_{-\infty}^{\infty} |w(u)|^2 du = \sigma_n^2\int_{-\infty}^{\infty} |W(f)|^2 df \qquad (2.24)$$

where $W(f)$ is the FT of $w(t)$. Also, from (2.15), b'_i can be written as

$$b'_i = \int_{-\infty}^{\infty} y(t_o+iT-u)w(u)du = \int_{-\infty}^{\infty} Y(f)\,W(f)e^{j2\pi f(t_o+iT)}df \qquad (2.25)$$

where $Y(f)$ is the FT of $y(t)$. In order to find the optimum $w(t)$ and $\{b_i\}$ according to the MMSE criterion, it is necessary to differentiate $\xi(N_1,\ N_2)$ with respect to $w(t)$, $\{b_i\}$ and set the result equal to zero. If $\xi(N_1,\ N_2)$ in (2.21) is differentiated with respect to b_i, one obtains

$$\delta\xi = -\sigma_s^2\sum_{\substack{i=-N_1\\ i\neq 0}}^{N_2} [(b'_i-b_i)\delta b_i^* + (b'_i-b_i)^*\delta b_i] \qquad (2.26)$$

Obviously $\delta\xi=0$ when

$$b_i = b'_i \qquad \text{for} \qquad -N_1\leq i\leq N_2\ , \quad i\neq 0 \qquad (2.27)$$

$\xi(N_1,\ N_2)$ now becomes

$$\xi(N_1,\ N_2) = \sigma_s^2\Big\{|b'_0-1|^2+\sum_{i=-\infty}^{-(N_1+1)} |b'_i|^2+\sum_{i=N_2+1}^{\infty} |b'_i|^2\Big\}+\sigma_n^2\int_{-\infty}^{\infty} |W(f)|^2 df \qquad (2.28)$$

Using (2.25), differentiating (2.28) with respect to $W(f)$ yields

$$\delta\xi = \int_{-\infty}^{\infty} 2\sigma_s^2 \, \mathrm{Re}\Big\{[-Y^*(f)e^{-j2\pi f t_o} +$$

$$Y^*(f)e^{-j2\pi f t_o}\int_{-\infty}^{\infty} Y(x)\,W(x)e^{j2\pi x t_o}\sum_{\substack{i>N_2\\ i=0\\ i<-N_1}} e^{j2\pi(x-f)iT}\,dx + \rho\,W(f)]\;\delta W^*(f)\Big\}\;df \tag{2.29}$$

where

$$\rho = \frac{\sigma_n^2}{\sigma_s^2} \tag{2.30}$$

For $\delta\xi = 0$ the integrand in (2.29) must be zero, i.e.

$$W(f) = Y^*(f)e^{-j2\pi f t_o}D(f) \tag{2.31}$$

where

$$D(f) = \frac{1}{\rho}\,[1-\int_{-\infty}^{\infty} Y(x)\,W(x)e^{j2\pi x t_o}\sum_{\substack{i>N_2\\ i=0\\ i<-N_1}} e^{j2\pi(x-f)iT}\,dx] \tag{2.32}$$

Note that $D(f)$ is periodic, with period T^{-1}, so that $W(f)$ in (2.31) can be thought of as the cascade of a continuous filter $Y^*(f)e^{-j2\pi f t_o}$ and a T-spaced transversal filter (i.e. periodic in frequency, with period T^{-1}) $D(f)$. $Y^*(f)e^{-j2\pi f t_o}$ is actually a filter "matched" to the impulse response $y(t)$, with a delay t_o. In fact the optimum linear receiving filter under various criteria can be expressed as the cascade of a matched filter and a transversal filter [22], [23]. Since $D(f)$ is a transversal filter, $W(f)$ followed by a sampler is equivalent to $Y^*(f)e^{-j2\pi f t_o}$ followed by a sampler followed by $D(f)$. Forney [23] has shown that the T-spaced samples (obtained at the correct sampling phase) of the output of a matched filter is a set of sufficient statistics for estimation of the transmitted data symbols. An explanation of this is given in Appendix A.

Substituting (2.31) into (2.32) yields

$$D(f) = \frac{1}{\rho}\,[1-\sum_k\int_{k/T}^{(k+1)/T} |Y(x)|^2 D(x)\sum_{\substack{i>N_2\\ i=0\\ i<-N_1}} e^{j2\pi(x-f)iT}\,dx]$$

$$= \frac{1}{\rho}\,[1-\sum_{\substack{i>N_2\\ i=0\\ i<-N_1}} e^{-j2\pi fiT}\, T\int_0^{T^{-1}} S_y(x)D(x)e^{j2\pi xiT}\,dx] \tag{2.33}$$

where
$$S_y(f) = \frac{1}{T}\sum_k |Y(f+\tfrac{k}{T})|^2 \tag{2.34}$$

$S_y(f)$ is real and periodic, with period T^{-1}. Similarly, substituting (2.31) into (2.24) and (2.25) yields

$$\sigma_{nw}^2 = \sigma_n^2 T\int_0^{T^{-1}} S_y(f)|D(f)|^2 df \tag{2.35}$$

$$b_i' = T\int_0^{T^{-1}} S_y(f)D(f)e^{j2\pi fiT}\,df \tag{2.36}$$

Since $S_y(f)D(f)$ has a period of T^{-1}, it can be expressed as

$$S_y(f)D(f) = \sum_k a_k e^{-j2\pi fkT} \tag{2.37}$$

where
$$a_k = T\int_0^{T^{-1}} S_y(f)D(f)e^{j2\pi fkT}\,df \tag{2.38}$$

$$\Rightarrow \quad b_i' = a_i \tag{2.39}$$

The optimum value of ξ, denoted as $\xi_{\min}(N_1, N_2)$, can be expressed as

$$\xi_{\min}(N_1, N_2) = \sigma_s^2\Big\{1-a_0-a_0^*+\sum_{\substack{i>N_2\\ i=0\\ i<-N_1}} |a_i|^2+\rho T\int_0^{T^{-1}} S_y(f)|D(f)|^2 df\Big\} \tag{2.40}$$

with $D(f)$ given by (2.33) and a_i given by (2.38). The quantities $D(f)$ and $\xi_{\min}(N_1, N_2)$ shall now be evaluated for combinations of N_1 and N_2.

2.2.1 Linear Equalizer (LE) [3], [21], [24], [25]

This is characterized by $N_1 = N_2 = 0$. The decision device then becomes a simple threshold detector. From (2.33) and (2.38),

$$D(f) = \frac{1}{\rho}[1-\sum_{i} e^{-j2\pi fiT} a_i]$$

$$= \frac{1}{\rho}[1-S_y(f)D(f)]$$

$$\Rightarrow \qquad D(f) = \frac{1}{(\rho+S_y(f))} \qquad \text{(real)} \tag{2.41}$$

Thus

$$W(f) = \frac{Y^*(f)e^{-j2\pi ft_o}}{(\rho+S_y(f))} \tag{2.42}$$

$$\xi_{\min}(0,\,0) = \sigma_s^2\Big\{1-a_0-a_0^*+\sum_i |a_i|^2+\rho T\int_0^{T^{-1}} S_y(f)|D(f)|^2 df\Big\}$$

$$= \sigma_s^2\Big\{1-2a_0+T\int_0^{T^{-1}} S_y^2(f)D^2(f)df + \rho T\int_0^{T^{-1}} S_y(f)D^2(f)df\Big\}$$

$$= \sigma_s^2\{1-a_0\} = \sigma_s^2 T\int_0^{T^{-1}} \frac{\rho}{(\rho+S_y(f))}df \tag{2.43}$$

2.2.2 Decision Feedback Equalizer (DFE) [3], [21], [24], [25]-[29]

This is characterized by $N_1=0$ and $N_2=\infty$. The decision device therefore subtracts the ISI contributions from all previous data symbols before providing an input to the threshold detector. From (2.33) and (2.38),

$$D(f) = \frac{1}{\rho}[1-\sum_{i=-\infty}^{0} e^{-j2\pi fiT} a_i] = \frac{1}{\rho}[1-[S_y(f)D(f)]^-] \tag{2.44}$$

where

$$[S_y(f)D(f)]^- \equiv \sum_{k=-\infty}^{0} a_k e^{-j2\pi fkT} \tag{2.45}$$

The impulse response of filter $D(f)$ is

$$d(t) = \sum_n d_n \delta(t-nT) \tag{2.46}$$

$$\Rightarrow \qquad D(f) = \sum_n d_n e^{-j2\pi fnT} \tag{2.47}$$

It is apparent from (2.44) that with $\rho\neq0$,

$$d_n = 0 \quad \text{for} \quad n > 0 \tag{2.48}$$

Therefore (2.44) can be written as

$$[(\rho+S_y(f))D(f)]^- = 1 \tag{2.49}$$

Since ρ and $S_y(f)$ are real, and with period T^{-1}, one can factorize $\rho+S_y(f)$ as

$$(\rho+S_y(f)) = M(f)M^*(f) \tag{2.50}$$

where
$$M(f) = \sum_{k=0}^{\infty} m_k e^{-j2\pi fkT} \tag{2.51}$$

Define
$$\Omega_M(z) = \sum_{k=0}^{\infty} m_k z^{-k} \tag{2.52}$$

If $M^*(f)^{-1}$ is to be expressable in the form

$$\frac{1}{M^*(f)} = \sum_{k=-\infty}^{0} \theta_k e^{-j2\pi fkT} \tag{2.53}$$

$$\theta_k \text{ finite as } k \rightarrow -\infty$$

then z-polynomial $\Omega_M(z)$ must be such that its roots lie on or inside the unit circle. The factorization of (2.50) can always be done so as to achieve this. Note that $\Omega_M(z)$ is the z-transform of the inverse FT of $M(f)$, and $M(f)=\Omega_M(e^{j2\pi fT})$. Note also that for $\rho>0$, $\Omega_M(z)$ has no roots on the unit circle, by virtue of (2.50) and the fact that $S_y(f)\geq0$. Substituting (2.50) into (2.49)

$$[M(f)M^*(f)D(f)]^- = 1 \tag{2.54}$$

The solution to (2.54) that satisfies (2.48) is

$$D(f) = \frac{1}{m_0 M^*(f)} \tag{2.55}$$

so that
$$W(f) = \frac{Y^*(f)e^{-j2\pi ft_o}}{m_0 M^*(f)} \tag{2.56}$$

Note that with $\rho=0$ there is no unique solution to $D(f)$. From (2.40),

$$\xi_{\min}(0,\infty) = \sigma_s^2\Big\{1-a_0-a_0^*+T\int_0^{T^{-1}} |[S_y(f)D(f)]^-|^2 df+\rho T\int_0^{T^{-1}} S_y(f)|D(f)|^2 df\Big\} \tag{2.57}$$

The quantity a_0 is the d.c. value of $S_y(f)D(f)$, and is given by

$$a_0 = 1-\frac{\rho}{|m_0|^2} \tag{2.58}$$

Using (2.44), eq. (2.57) becomes

$$\begin{aligned}\xi_{\min}(0,\infty) &= \sigma_s^2\Big\{1-2(1-\frac{\rho}{|m_0|^2})+1-\frac{2\rho}{|m_0|^2}+\rho T\int_0^{T^{-1}} (\rho+S_y(f))|D(f)|^2 df\Big\}\\ &= \frac{\sigma_s^2\rho}{|m_0|^2}\end{aligned} \tag{2.59}$$

From (2.50),

$$\ln(\rho+S_y(f)) = \ln M(f)+\ln M^*(f) \tag{2.60}$$

Since $\Omega_M(z)$ has all zeros on or inside the unit circle, $\ln M(f)$ is of the form

$$\ln M(f) = \ln m_0+\sum_{k=1}^{\infty}\theta_k e^{-j2\pi fkT} \tag{2.61}$$

Thus

$$\ln|m_0|^2 = T\int_0^{T^{-1}} \ln[\rho+S_y(f)]\, df \tag{2.62}$$

giving

$$\xi_{\min}(0,\infty) = \sigma_s^2\, e^{\{-T\int_0^{T^{-1}} \ln[\frac{\rho+S_y(f)}{\rho}]df\}} \tag{2.63}$$

2.2.3 Matched Filter Equalizer (MFE) [25]

The MFE is characterized by $N_1=\infty$ and $N_2=\infty$. The decision device is able to subtract ISI from both future and past data symbols before providing an input to the threshold detector. From (2.33),

$$D(f) = \frac{1}{\rho}[1-T\int_0^{T^{-1}} S_y(x)D(x)dx] \tag{2.64}$$

Using the form for $D(f)$ given in (2.47), one can see that

$$D(f) = d_0 \tag{2.65}$$

Substituting (2.65) in (2.64),

$$d_0 = \left(\rho + T\int_0^{T^{-1}} S_y(x)dx\right)^{-1} = \frac{1}{(\rho+E_n)} \tag{2.66}$$

where $$E_n = T\int_0^{T^{-1}} S_y(f)df = \int_{-\infty}^{\infty} |Y(f)|^2 df = \int_{-\infty}^{\infty} |y(t)|^2 dt \tag{2.67}$$

The quantity E_n is the "energy" of the received impulse response $y(t)$. The filter $W(f)$ is given by

$$W(f) = \frac{Y^*(f)e^{-j2\pi f t_o}}{(\rho+E_n)} \tag{2.68}$$

From (2.40),

$$\xi_{\text{min}}(\infty,\, \infty) = \sigma_s^2\left\{1 - \frac{2E_n}{(\rho+E_n)} + \frac{E_n^2}{(\rho+E_n)^2} + \frac{\rho E_n}{(\rho+E_n)^2}\right\} = \frac{\sigma_s^2 \rho}{(\rho+E_n)} \tag{2.69}$$

It can be seen that $\xi_{\text{min}}(0,\, 0) \geq \xi_{\text{min}}(0,\, \infty) \geq \xi_{\text{min}}(\infty,\, \infty)$. Before talking more about the relative merits of the three MMSE equalizers just described, the optimum detector (in the sense of minimum probability of error) and the maximum likelihood sequence estimator (MLSE) will be discussed.

2.3 Optimum Detection and the MLSE

From the QAM baseband model of fig. 2.3, the received signal, $r(t)$, is given by

$$r(t) = \sum_i s_i y(t - iT) + n(t) \tag{2.70}$$

Suppose that the receiver observes the signal $r(t)$ over the time interval $\mathfrak{I}$, corresponding to the transmitted data sequence s_0, s_1, ..., s_I, which we represent as $\{s_i\}_0^I$. The receiver would like to make an estimate of $\{s_i\}_0^I$, denoted as $\{\hat{s}_i\}_0^I$, with the minimum probability of error. This can be achieved by choosing $\{\hat{s}_i\}_0^I$ out of the $L^{2(I+1)}$ possible sequences to be such that

$$\Pr[\{\hat{s}_i\}_0^I \,/\, r(t), t \in \mathfrak{I}] \quad \text{is maximized} \tag{2.71}$$

This is known as the maximum a-posteriori (MAP) criterion. If $\{s_i\}_0^I$ is equally likely to take any one of its $L^{2(I+1)}$ possible forms, then the criterion of selecting the estimate $\{\hat{s}_i\}_0^I$ to be such that

$$\Pr[r(t), t \in \Im \;/\{\hat{s}_i\}_0^I] \quad \text{is maximized} \tag{2.72}$$

also achieves minimum probability of error. This is known as the maximum likelihood (ML) criterion, and is only equivalent to MAP under the assumption of equally likely data sequences. Of course, equally likely data sequences is assured if the data symbols are independent and equally likely to take any one of their L^2 possible values. In terms of the probability distribution function, the ML criterion can be expressed as selecting $\{\hat{s}_i\}_0^I$ such that

$$p[r(t), t \in \Im \;/\{\hat{s}_i\}_0^I] \quad \text{is maximized} \tag{2.73}$$

Now using (B.28), (B.29) and (B.30) of Appendix B, and remembering that $n(t)$ is a stationary complex zero mean white gaussian random process, with independent real and imaginary parts each having the statistical autocorrelation function $\sigma_n^2 \delta(\tau)/2$, one can write

$$\begin{aligned} p[r(t), t \in \Im \;/\{s_i\}_0^I] &= p[n(t), t \in \Im \;/\{s_i\}_0^I] \\ &= p[n(t), t \in \Im] \sim \mathrm{e}^{\{-\int_{\Im} |n(t)|^2 dt/\sigma_n^2\}} \end{aligned} \tag{2.74}$$

Substituting for $n(t)$ from (2.70), with the summation going from 0 to I, and collecting only terms that depend on $\{s_i\}$, gives

$$p[r(t), t \in \Im \;/\{s_i\}_0^I] \sim \mathrm{e}^{\{\sum_{i=0}^{I} 2\mathrm{Re}[s_i^* v_i] - \sum_{i=0}^{I} \sum_{m=0}^{I} s_i^* l_{i-m} s_m\}/\sigma_n^2} \tag{2.75}$$

where
$$v_i = \int_{\Im} r(t) y^*(t - iT) dt \tag{2.76}$$

$$l_{i-m} = \int_{\Im} y(t - mT) y^*(t - iT) dt = l_{m-i}^* \quad \text{for} \quad I \geq (i,\, m) \geq 0 \tag{2.77}$$

Remembering that $y(t)$ is causal, and assuming that the period of observation $\Im$ at least covers

0 to (IT+time spread of $y(t)$), then (2.76) and (2.77) can be rewritten as

$$v_i = \int_{-\infty}^{\infty} r(t)y^*(t-iT)dt \tag{2.78}$$

$$l_{i-m} = \int_{-\infty}^{\infty} y(t-mT)y^*(t-iT)dt = l^*_{m-i} \quad \text{for} \quad I\geq(i,\ m)\geq 0 \tag{2.79}$$

$\{v_i\}$ is recognizable as the samples of the output of a matched filter, whilst $\{l_i\}$ are the samples of the impulse response of the cascade of the channel $y(t)$ and matched filter $y^*(-t)$, or, in other words, the autocorrelation function of $y(t)$. Let an infinite sequence of symbols $\{\Upsilon_i\}$ be such that

$$\Upsilon_i = \text{a non-zero data symbol value for } 0\leq i\leq I\text{, and 0 otherwise} \tag{2.80}$$

If $\{\Upsilon_i\}^N$ denotes the sequence from $-\infty$ to N, then define the metric for this sequence as

$$J(\{\Upsilon_i\}^N) = \sum_{i=\infty}^{N} 2\text{Re}[\Upsilon_i^* v_i] - \sum_{i=-\infty}^{N} \sum_{m=-\infty}^{N} \Upsilon_i^* l_{i-m} \Upsilon_m$$

$$= \sum_{i=0}^{N}\Big(2\text{Re}[\Upsilon_i^* v_i] - \sum_{m=0}^{N} \Upsilon_i^* l_{i-m} \Upsilon_m\Big) \tag{2.81}$$

Then from (2.75), for the estimate $\{\hat{s}_i\}_0^I$ of $\{s_i\}_0^I$ to satisfy the ML criterion, $\{\hat{s}_i\}_0^I$ must be such that over all $L^{2(I+1)}$ possible sequences $\{\Upsilon_i\}^I$ the metric $J(\{0\}^{-1}\{\hat{s}_i\}_0^I)$ is maximum, where $\{0\}^{-1}$ denotes a semi-infinite sequence of zeros from $-\infty$ to -1. In view of the length and number of possible sequences, the brute-force approach of calculating $J(.)$ for all sequences and selecting that which yields a maximum value is impractical. However, computational effort can be greatly reduced, while still achieving ML detection, through the use of a recursive algorithm based on the principles of dynamic programming and known as the Viterbi Algorithm (VA). A modified version of the VA [30] will now be described.

2.3.1 Implementation of the MLSE by the VA

It is possible to write $J(.)$ recursively as

$$J(\{\Upsilon_i\}^N) = J(\{\Upsilon_i\}^{N-1})+\mathrm{Re}[\Upsilon_N^*(2v_N-\Upsilon_N l_0-2\sum_{i=0}^{N-1}\Upsilon_i l_{N-i})] \tag{2.82}$$

Let integer g be such that

$$y(t)=0 \text{ for } 0\leq t<(g+1)T \tag{2.83}$$

$$\Rightarrow \quad y(iT) = 0 \quad \text{for } i<0,\ i>g \tag{2.84}$$

$$l_i = 0 \quad \text{for } |i|>g \tag{2.85}$$

Then (2.82) becomes

$$J(\{\Upsilon_i\}^N) = J(\{\Upsilon_i\}^{N-1})+\mathrm{Re}[\Upsilon_N^*(2v_N-\Upsilon_N l_0-2\sum_{i=1}^{g}\Upsilon_{N-i} l_i)] \tag{2.86}$$

It can be seen that in the evaluation of metric $J(\{\Upsilon_i\}^N)$, the second term on the RHS of (2.86) involves the last g symbols of $\{\Upsilon_i\}^{N-1}$. Suppose that the g symbols $\Upsilon_{N-g}, \Upsilon_{N-g+1}, \dots, \Upsilon_{N-1}$ are known, then from (2.86) the sequence $\{\Upsilon_i\}^I$ with the maximum metric also has the subsequence $\{\Upsilon_i\}^{N-1}$, $N\leq I$, with the maximum metric. Of course the last g symbols of $\{\Upsilon_i\}^{N-1}$ are not known to the receiver, but they do, at most, take on L^{2g} different forms. Thus the receiver needs to store, at most, only L^{2g} sequences at any one time. The algorithm can be broadly stated as follows.

Assume $I\geq g$, and note that for a sequence $\{\Upsilon_i\}^N$, $0\leq N\leq I$, the receiver need only store the non-zero part, i.e. $\{\Upsilon_i\}_0^N$. At time $0T$, but before the receipt of sample v_0, the receiver holds no sequences. On receipt of v_0 the receiver stores the L^2 sequences $\{\Upsilon_i\}^0$, which is just all the different possible data symbols, together with their respective metrics as given by (2.81). At time NT, $0<N<g$, but before the receipt of sample v_N, the receiver holds in store the L^{2N} different possible sequences $\{\Upsilon_i\}^{N-1}$ together with their respective metrics. On receipt of v_N each stored sequence $\{\Upsilon_i\}^{N-1}$ is expanded into L^2 sequences $\{\Upsilon_i\}^N$, the last symbol of each expanded sequence taking exclusively one of its L^2 possible values. The metrics of the new expanded sequences are evaluated according to (2.86). The receiver now has $L^{2(N+1)}$ sequences $\{\Upsilon_i\}^N$ with their respective metrics. At time NT, $g\leq N\leq I$, but before the receipt of sample v_N, the receiver holds in store L^{2g} sequences $\{\Upsilon_i\}^{N-1}$ together with their respective metrics.

Each of the sequences differs from the other at least in its last g symbols. On receipt of v_N each stored sequence is expanded and the metric updated, according to (2.86), as previously. The receiver now has $L^{2(g+1)}$ sequences $\{\Upsilon_i\}^N$ with their respective metrics, and for all sequences with the same last g symbols the receiver proceeds to extract that with the largest metric and discard the rest. Thus the receiver finishes up with L^{2g} sequences $\{\Upsilon_i\}^N$ (called survivors) and their respective metrics. After the receipt of sample v_I the ML estimate $\{\hat{s}_i\}_0^I$ is given by the non-zero part of that sequence $\{\Upsilon_i\}^I$ with the largest metric. To prevent the metrics from growing excessively large with time, it is a good idea to periodically subtract the smallest metric from all the metrics.

The modified form of the VA just described differs from the original in that it operates on samples of the output of a matched filter, where the noise samples are in general correlated. The original VA [3], [21], [23], [31] requires noise samples that are statistically independent. The form of the original VA can be derived from (2.81) as follows.

The quantities v_i (2.78) and l_{i-m} (2.79) can be written in terms of FT's as

$$v_i = \int_{-\infty}^{\infty} R(f)\,Y^*(f)\,e^{j2\pi f iT}\,df \tag{2.87}$$

$$l_{i-m} = \int_{-\infty}^{\infty} |Y(f)|^2 e^{j2\pi f(i-m)T}\,df = T\int_{0}^{T^{-1}} S_y(f)\,e^{j2\pi f(i-m)T}\,df \tag{2.88}$$

where $R(f)$ is the FT of $r(t)$. It is apparent form (2.88) and (2.85) that

$$S_y(f) = \sum_{i=-g}^{g} l_i e^{-j2\pi f iT} \tag{2.89}$$

Since $S_y(f)$ is real and periodic, it can factorized as

$$S_y(f) = M'(f)M'^*(f) \tag{2.90}$$

where
$$M'(f) = \sum_{i=0}^{g} m'_i e^{-j2\pi f iT} \tag{2.91}$$

From (2.81),

$$J(\{\Upsilon_i\}^I) = 2\mathrm{Re}[\int_{-\infty}^{\infty} \frac{R(f)Y^*(f)}{M'^*(f)} (\sum_{i=0}^{I} \Upsilon_i^* M'^*(f) e^{j2\pi fiT}) df] - $$

$$T\int_0^{T^{-1}} |\sum_{i=0}^{I} \Upsilon_i M'(f) e^{-j2\pi fiT}|^2 df \tag{2.92}$$

Note that the ratio $Y^*(f)/M'^*(f)$ in the above expression is well defined as $M'(f)$, by definition, is such that

$$\sum_i \frac{|Y(f+\frac{i}{T})|^2}{|M'(f+\frac{i}{T})|^2} = T \tag{2.93}$$

Now
$$\sum_{i=0}^{I} \Upsilon_i M'(f) e^{-j2\pi fiT} = \sum_{i=0}^{I+g} Z_i' e^{-j2\pi fiT} \tag{2.94}$$

where
$$Z_i' = \sum_{h=0}^{g} \Upsilon_{i-h} m_h' \tag{2.95}$$

Let
$$Z_i = \int_{-\infty}^{\infty} \frac{R(f)Y^*(f)}{M'^*(f)} e^{j2\pi fiT} df \tag{2.96}$$

Thus
$$J(\{\Upsilon_i\}^I) = \mathrm{Re}[\sum_{i=0}^{I+g} Z_i'^*(2Z_i - Z_i')] \tag{2.97}$$

The quantity Z_i is recognizable as the sample at iT of $r(t)$ passed through the filter $Y^*(f)/M'^*(f)$, while Z_i' is a possible value of the data portion of the sample Z_i. Note that the sampled impulse response of $|Y^*(f)|^2/M'^*(f)$ is causal with FT given by $M'(f)$. If $\tilde{n}_k$ and $\tilde{n}_i$ are samples at times kT and iT respectively of the noise $n(t)$ passed through the filter $Y^*(f)/M'^*(f)$, it is quite easy to show that

$$\mathrm{E}[\tilde{n}_k \tilde{n}_i^*] = \sigma_n^2 \int_0^{T^{-1}} \sum_h \frac{|Y(f+\frac{h}{T})|^2}{|M'(f+\frac{h}{T})|^2} e^{j2\pi f(k-i)T} df = \sigma_n^2 \delta_{ki} \tag{2.98}$$

where δ_{ki} is the Kronecker delta function,

$$\begin{aligned} \delta_{ki} &= 1 \quad \text{for } k{=}i \\ &= 0 \quad \text{for } k{\neq}i \end{aligned} \tag{2.99}$$

Thus the noise samples are uncorrelated, or equivalently, statistically independent since the noise process is gaussian. The filter $Y^*(f)/M'^*(f)$ is identical to the "whitened matched filter" of [23], [32], so-called because the noise samples at its output are statistically independent. Define for the sequence $\{\Upsilon_i\}^N$ the metric

$$\Gamma(\{\Upsilon_i\}^N) = \mathrm{Re}[\sum_{i=0}^{N} Z_i'^*(2Z_i - Z_i') \tag{2.100}$$

Note that

$$J(\{\Upsilon_i\}^I) = \Gamma(\{\Upsilon_i\}^{I+g}) \tag{2.101}$$

so that the ML estimate $\{\hat{s}_i\}_0^I$ of $\{s_i\}_0^I$ is such that over all $L^{2(I+1)}$ possible sequences $\{\Upsilon_i\}^{I+g}$ the metric $\Gamma(\{0\}^{-1}\{\hat{s}_i\}_0^I\{0\}_{I+1}^{I+g})$ is maximum. $\Gamma(.)$ can be written recursively as

$$\Gamma(\{\Upsilon_i\}^N) = \Gamma(\{\Upsilon_i\}^{N-1}) + \mathrm{Re}[Z_N'^*(2Z_N - Z_N')] \tag{2.102}$$

From (2.95) it can be seen that Z_N' involves the last g symbols of $\{\Upsilon_i\}^{N-1}$ so that, as before, the receiver need only store at most L^{2g} sequences at any one time. The original VA, up to the receipt of sample Z_I at time IT, proceeds exactly as in the modified VA described previously except that now we are talking about samples $\{Z_i\}$ and referring to the metric $\Gamma(.)$. After the receipt of Z_I the receiver holds L^{2g} sequences $\{\Upsilon_i\}^I$ with their respective metrics, but the ML estimate is not now given by the non-zero part of that sequence with the largest metric. At time NT, $I<N<I+g$, but before the receipt of sample Z_N, the receiver holds $L^{2(g+I+1-N)}$ sequences $\{\Upsilon_i\}^I$ with their metrics. Each of the sequences differs from the other at least in its last $g+I+1-N$ symbols. On receipt of Z_N the metric for each sequence is updated according to (2.102), with no expansion being performed, and then for all sequences with the same last $g+I-N$ symbols the receiver extracts that with the largest metric and discards the rest. Thus the receiver finishes up with $L^{2(g+I-N)}$ sequences $\{\Upsilon_i\}^I$ with their metrics. At time $(I+g)T$, but before the receipt of sample Z_{I+g}, the receiver holds L^2 sequences $\{\Upsilon_i\}^I$ with their metrics. On receipt of Z_{I+g} the metrics are updated, and now the ML estimate $\{\hat{s}_i\}_0^I$ is given by the non-zero part of that sequence $\{\Upsilon_i\}^I$ with the largest metric.

This completes the description of the VA. The original form tends to be the more widely used version. Symbol errors in the VA tend to occur in bursts, the lengths, for a given channel response, being variable as they depend on the noise samples and data symbols that are present at the time. Bounds on the VA's probability of error performance can be found in [3], [23], [30], [33]. Error bounds for the original VA in the presence of correlated noise samples are developed in [34].

The modified version of the VA can be adapted to deal with the case of non-white gaussian noise as shown in [30]. The VA performs about $(k+1)L^{2(g+1)}$ metric computations in the detection of $k+1$ transmitted symbols, compared with $L^{2(k+1)}$ needed for the brute-force method. A problem, however, is that the stored sequences grow linearly with time. For a large k this can lead to storage problems, and so in practice the sequences are truncated to some length q $(q \gg g)$. Then, say, at time NT on processing the N^{th} received sample, the receiver selects as the detected symbol $\hat{s}_{N-q}$ that Υ_{N-q} belonging to the sequence with the largest metric, and finishes with L^{2g} stored sequences $\{\Upsilon_i\}_{N-q+1}^{N}$. Thus there is a delay in detection of qT. The loss in performance resulting from this truncation strategy is negligible if $q \geq 5g$ [3], as most stored sequences will by then have a common sub-sequence $\{\Upsilon_i\}^{N-q}$. For large constellations and/or a large value of g, storage can again be a problem as well as number of computations. In such cases one strategy is to use a transversal pre-filter to reduce the value of g seen by the VA to some desired setting [24], [34]-[37], although this enhances the noise and makes the samples, if uncorrelated, correlated. Another strategy is to reduce the number of sequences stored by the detector [37]-[43], since many sequences are so unlikely to be the maximum likelihood sequence that they can effectively be ignored. These *reduced state* detectors are usually referred to as *near-MLSE's*, and in the literature almost invariably take the form of some sub-optimum version of the original VA. There is usually an increase in the number of computations per stored sequence for the near-MLSE. For best performance, the near-MLSE usually requires that the impulse response seen by the algorithm, i.e.

$$\sum_{h=0}^{g} m'_h \delta(t-hT)$$

have its "energy" concentrated towards the beginning of the response, i.e. be *minimum phase.* This is because if, for example, the first component m_0' in the impulse response is small relative to the rest, then on processing of sample Z_i at time iT, the transmitted symbol s_i plays a less effective role in the selection of which sequences to keep, and therefore also which possible values it will be detected as, and which sequences to discard. The minimum phase condition can be achieved if the z-polynomial

$$\Omega_{M'}(z) = \sum_{h=0}^{g} m_h' z^{-h} \tag{2.103}$$

i.e. the z-transform of the impulse response seen by the near-MLSE algorithm, has its roots on or inside the unit circle. Then the resulting receiver filter $Y^*(f)/M'^*(f)$ becomes identical (except for a scaling factor) to the optimum filter for the ZF DFE, a result first shown in [44]. Compare $M(f)$ for the MMSE DFE (2.50) with (2.90); it is apparent that

$$M(f)M^*(f) = M'(f)M'^*(f)+\rho \tag{2.104}$$

It can be seen that for small ρ, the receiver filter (2.56) for the MMSE DFE approximates to that for the ZF DFE.

2.3.2 Lower Bound to Performance

Any receiver can perform no better than if it were able to detect each transmitted data symbol in total isolation, so that there would be no ISI. For a white gaussian noise channel, setting $I=0$ in (2.75) gives

$$p[r(t),t\in\mathfrak{I}\ /s_0] \sim e^{\{2\mathrm{Re}[s_0^* v_0]-|s_0|^2 l_0\}/\sigma_n^2} \sim e^{\{-l_0|(v_0/l_0)-s_0|^2\}/\sigma_n^2} \tag{2.105}$$

Thus $\hat{s}_0$ is given by that symbol Υ_0 that minimizes $|(v_0/l_0)-\Upsilon_0|$ over all L^2 possible data symbols. This is equivalent to threshold detecting from sample v_0/l_0. The receiver filter is $Y^*(f)/E_n$, where E_n is given by (2.67) ($E_n=l_0$). This is just the ZF matched filter equalizer.

2.4 Comparison

Table 2.1 gives a summary of the receiver designs discussed so far, with reference to the transmission model of fig. 2.3.

Design	Filter *W(f)*	Decision Device
MMSE LE	$\dfrac{Y^*(f)e^{-j2\pi ft_o}}{(\rho+S_y(f))}$	Threshold detector
MMSE DFE	$\dfrac{Y^*(f)e^{-j2\pi ft_o}}{m_0M^*(f)}$ $\Omega_M(z)$ has its roots on or inside the unit circle.	Threshold detector, with ISI cancellation for previous symbols.
MMSE MFE	$\dfrac{Y^*(f)e^{-j2\pi ft_o}}{(\rho+E_n)}$	Threshold detector, with total ISI cancellation.
ZF MFE	$\dfrac{Y^*(f)e^{-j2\pi ft_o}}{E_n}$	Threshold detector, with total ISI cancellation.
MLSE/near-MLSE	$\dfrac{Y^*(f)e^{-j2\pi ft_o}}{M'^*(f)}$ For near-MLSE, $\Omega_{M'}(z)$ has its roots on or inside the unit circle.	VA (original), or some reduced-state version.

Table 2.1 Receiver designs for different equalizers.

If the channel spectrum *Y(f)* has nulls or points of very low amplitude, then, depending on the function $S_y(f)$, the corresponding receiver filter *W(f)* for the LE may have points of very high amplitude, leading to excessive noise enhancement [24]. This is not so for the DFE, which in general, therefore, has a significantly superior error rate performance [3]. For a time-varying channel such as HF, there are bound to be many instances where nulls or near nulls appear in the spectrum characteristic, and so the LE is not recommended. Furthermore, the DFE is much more insensitive than the LE to errors in the receiver's knowledge of the carrier phase ϕ and the sampling phase t_o [24], [45], [46]. It has not been mentioned yet but, as the name suggests, the DFE uses previous decisions for the ISI cancellation. Obviously if decisions are incorrect then error propagative effects arise. These effects are difficult to analyse quantitively without resorting to Monte Carlo simulation on a computer. Some analytical results on the probability of error of the DFE in the presence of decision errors can be found in [29], [47].

The MFE can be implemented in practice, when it is then sometimes referred to as a decision-aided ISI canceller [24], by using tentative decisions regarding future data symbols. A receiver with a two-step decision process such as this was first proposed in [48]. Like the DFE, this receiver will suffer from propagative effects due to incorrect decisions. We are interested in the MFE only as an indicator of the lower bound to performance, the ZF MFE in the absence of decision errors achieving this. Comparison of the ZF MFE with the MMSE MFE illustrates the fact that minimization of MSE does not necessarily mean minimization of probability of error. A DFE based on the minimization of probability of error has been designed (see [29] and the references therein) and can be shown to be equivalent to the MMSE DFE for high signal-to-noise ratios. Also, at high signal-to-noise ratios the MLSE approaches the lower bound to performance for all channels except those with extremely severe ISI [23]. For $g=0$ (no ISI) the MMSE DFE becomes equivalent to the MMSE MFE while the MLSE and ZF DFE become equivalent to the ZF MFE.

In making a decision on a transmitted data symbol, the MLSE processes all the energy in the sampled impulse response of the channel in cascade with the receiver filter, i.e. $Y(f)\,W(f)$, whereas the DFE cancels out those terms of the impulse response that give rise to ISI (assuming past decisions are correct). For this reason the MLSE usually outperforms, in terms of probability of error, the DFE, e.g. [3], [21], [27], [29], [45], the trade-off being complexity. Note that it is assumed the receiver knows perfectly the channel response $y(t)$ and sampling phase t_o. In [49] it is shown that the DFE can outperform the MLSE, implemented by the VA, in the presence of phase jitter.

2.5 Fractionally-Spaced Transversal Filters

Table 2.1 shows that the receiver filter $W(f)$ for all the designs can be expressed as the cascade of a matched filter $Y^*(f)e^{-j2\pi ft_o}$ and a T-spaced transversal filter, denoted as $D(f)$. Linearity permits us to replace $W(f)$ followed by a sampler by $Y^*(f)e^{-j2\pi ft_o}$ followed by a sampler followed by $D(f)$. Then $D(f)$ can be implemented as a digital non-recursive filter as shown in fig. 2.4a. The continuous filter $Y^*(f)e^{-j2\pi ft_o}$ is dependent on the sampling phase t_o, which according to [3] should be known to within ±5% of T for most applications. Techniques for sampling with t_o at some desired setting (usually zero), termed *synchronization*, can be found in [3], [50]-[55]. QAM has an advantage over PAM systems in that timing with significantly less jitter can be obtained [3], [50]. The penalty suffered, in terms of MSE, for inaccuracies in the knowledge of t_o for LE's and DFE's is demonstrated in [24]. Also, as indicated in [24], the relative insensitivity of the DFE over the LE for inaccuracies in t_o may be offset by an increased severity in error propagation when a decision error occurs. The problem with continuous filters is that it is difficult to make them adaptive to change, whereas, as we shall see later, digital filters can be made adaptive relatively easily as they only require a finite number of coefficients (or taps) to be updated. Thus a receiver filter with a fixed continuous filter, which ideally should be $Y^*(f)e^{-j2\pi ft_o}$ at all times, is going to suffer not just because of inaccuracies in t_o but also, and more seriously, if the channel is time-varying. What is desired is a receiver filter that is equivalent to $W(f)$ but which consists of continuous filters that can remain fixed and transversal filters that can be made adaptive.

Let the bandwidth of $Y(f)$ be denoted as f_y, so that

$$Y(f) = 0 \qquad \text{for } |f| > \frac{f_y}{2} \tag{2.106}$$

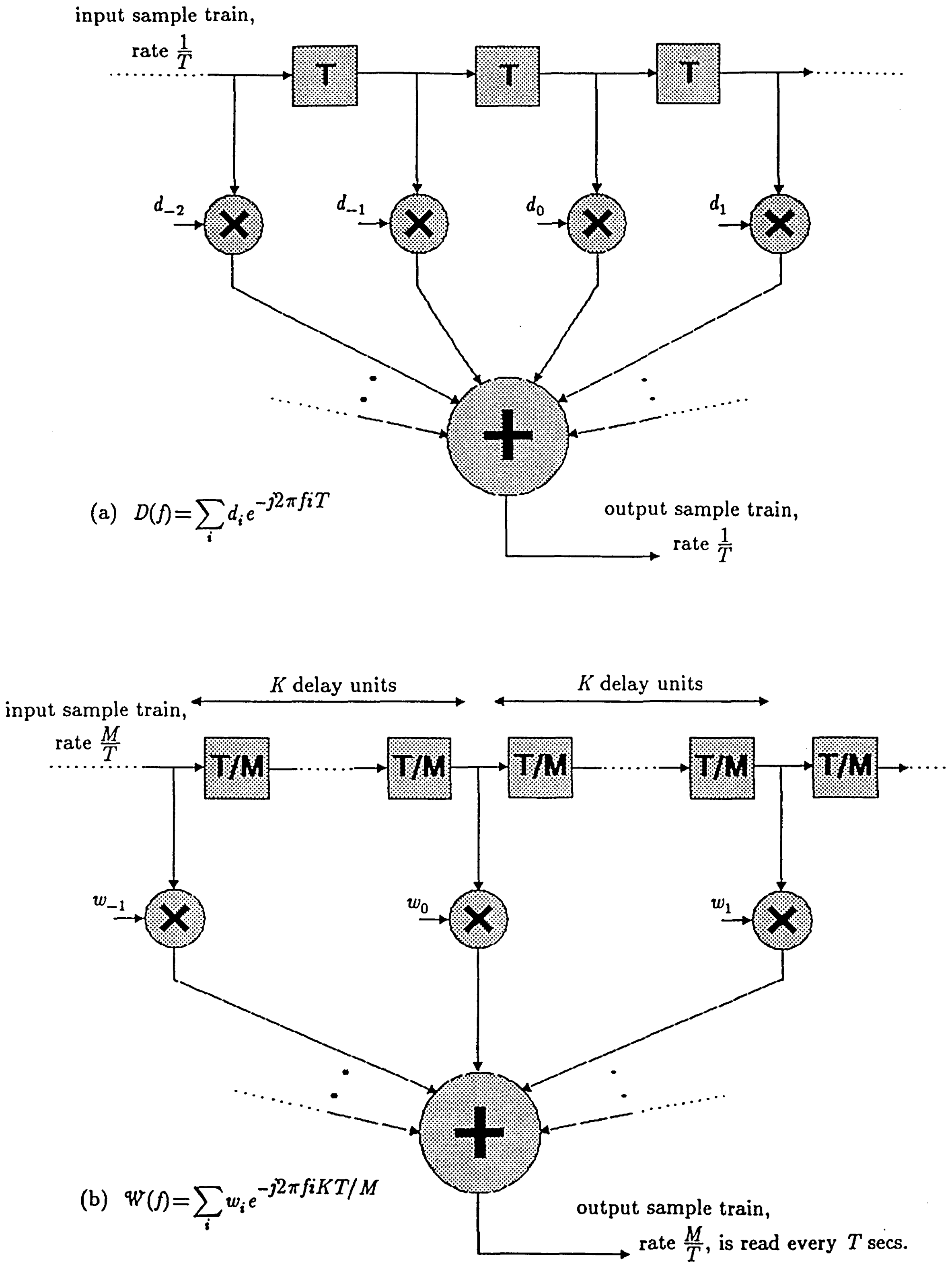


Fig. 2.4 Digital implementation of (a) T – spaced transversal filter $D(f)$, (b) KT/M fractionally-spaced filter $\mathcal{W}(f)$.

The bandwidth of $W(f)$ will also by f_y. Let filter $\mathcal{B}(f)$ have bandwidth $\mathcal{T}^{-1}$ and be such that

$$\mathcal{B}(f) \neq 0 \quad \text{for } |f| \leq \frac{1}{2\mathcal{T}} \tag{2.107}$$

Assume $\mathcal{T}^{-1} > f_y$, and let the $\mathcal{T}$-spaced transversal filter $\mathcal{W}(f)$ be given by

$$\mathcal{W}(f) = \sum_i \frac{W(f+i/\mathcal{T})}{\mathcal{B}(f+i/\mathcal{T})} \tag{2.108}$$

Then the filter $\mathcal{B}(f)$ followed by $\mathcal{W}(f)$ is equivalent to $W(f)$. To implement $\mathcal{W}(f)$ as a digital filter we sample the output of the continuous filter $\mathcal{B}(f)$ at rate M/T (M is an integer ≥ 1), the samples being shifted into a digital shift register memory. Then every K^{th} sample ($K \geq 1$) in the shift register is multiplied by a successive filter coefficient (or tap) and all the products summed to give an output that is read at rate T^{-1}. Thus $\mathcal{T} = KT/M$, where K, M are relative prime integers. If $f_y \leq T^{-1}$ then $K = M = 1$ is sufficient, and if $f_y > T^{-1}$, as often happens in practice, we need $K < M$. The filter $\mathcal{W}(f)$ is termed a *fractionally-spaced transversal filter* because $\mathcal{T}$ is generally required to be less than T. The digital implementation of $\mathcal{W}(f)$ is illustrated in fig. 2.4b. The filter $\mathcal{B}(f)$ can be any fixed continuous filter satisfying (2.107), the obvious one coming to mind having a rectangular spectrum with cut-off at $\pm 0.5/\mathcal{T}$. For more details on fractionally-spaced transversal filters see [24] and the references therein.

In this thesis it will henceforth be assumed that $f_y \lesssim T^{-1}$, a condition guaranteed if the bandwidth of the pulse shaping filter $a(t)$ is $\lesssim T^{-1}$ (recall (2.13)). For a voiceband HF channel we would want the bandwidth of $a(t)$ to be as close to T^{-1} as possible. Thus we shall assume that effectively $\mathcal{T} = T$, and the receiver filter consists of a fixed filter $\mathcal{B}(f)$, which will be assumed to be rectangular with amplitude $\sqrt{T}$ and cut-off at $\pm 0.5/T$, followed by a T-spaced transversal filter $\mathcal{W}(f)$ implemented as a digital non-recursive filter. The receiver arrangement is shown in fig. 2.5. The noise samples at the output of $\mathcal{B}(f)$ will be uncorrelated. The transversal filter for the MLSE is a pure phase filter, i.e. constant amplitude spectrum, because $|\mathcal{W}(f)|^2 = 1$, as is also that for the MMSE DFE at high signal-to-noise ratios. Note that for a MLSE, as opposed to a near-MLSE, the transversal filter can be omitted as we are free to

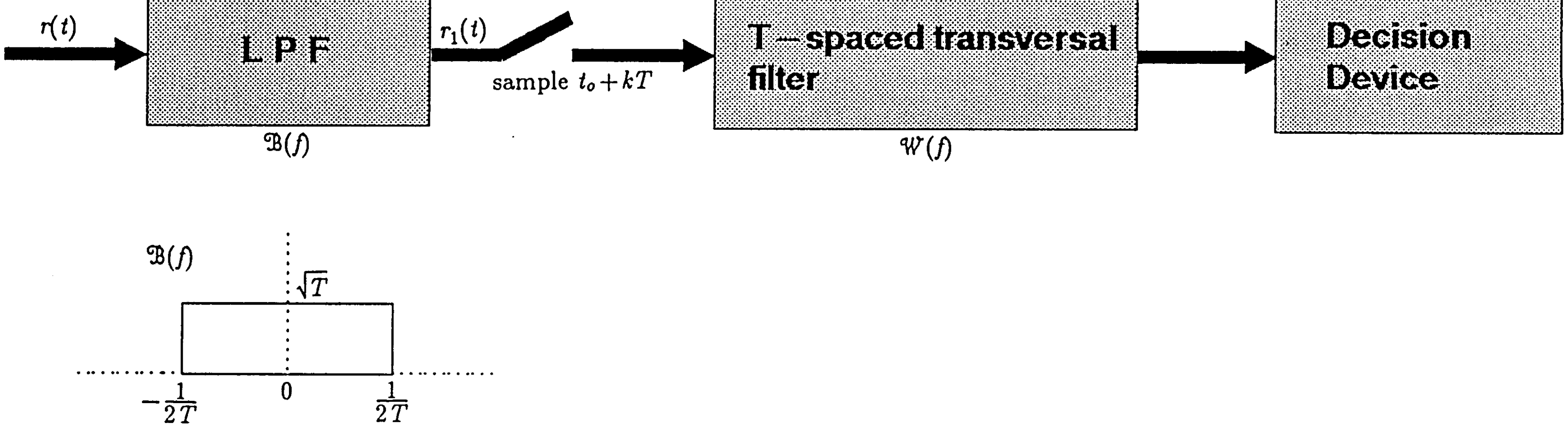


Fig. 2.5 Receiver arrangement for $f_y \leq \frac{1}{T}$ (c.f. fig. 2.3).

choose $M'(f)=\sqrt{T}^{-1}\sum_i Y(f+\frac{i}{T})e^{j2\pi(f+\frac{i}{T})t_o}$ so that $\mathcal{W}(f)=1$.

The sample at time t_o+kT of the output $r_1(t)$ of filter $\mathcal{B}(f)$, which has impulse response $b_1(t)$, is

$$r_1(t_o+kT) = \sqrt{T}\sum_i s_i y(t_o+(k-i)T)+n_1(t_o+kT) \tag{2.109}$$

where

$$n_1(t) = n(t)\bullet b_1(t) \tag{2.110}$$

Replacing t_o by $HT+\Delta$, where H is an integer and $0\leq\Delta<T$, gives

$$r_1((k+H)T+\Delta) = \sqrt{T}\sum_i s_i y((k+H-i)T+\Delta)+n_1((k+H)T+\Delta) \tag{2.111}$$

Define

$$r_i = r_1(iT+\Delta) \tag{2.112}$$

$$y_i = \sqrt{T}\ y(iT+\Delta) \tag{2.113}$$

$$n_i = n_1(iT+\Delta) \tag{2.114}$$

From (2.83), $y_i=0$ for $i<0$, $i>g$. Therefore (2.111) becomes

$$r_{k+H} = \sum_{i=0}^{g} s_{k+H-i}y_i+n_{k+H} \tag{2.115}$$

Note that

$$\mathrm{E}[n_i n_k^*] = \sigma_n^2\delta_{ik} \tag{2.116}$$

The noise samples at the input to $\mathcal{W}(f)$ are therefore statistically independent, i.e. white noise samples. The z-transform of the sampled waveform of $\sqrt{T}y(t+\Delta)$, with zero sampling phase, is

$$\Omega_Y(z) = \sum_{i=0}^{g} y_i z^{-i} \tag{2.117}$$

Also,

$$\Omega_Y(e^{j2\pi fT}) = \frac{1}{\sqrt{T}}\sum_i Y(f+\frac{i}{T})e^{j2\pi(f+\frac{i}{T})\Delta} \tag{2.118}$$

Note that $\Omega_Y(z)\Omega_Y^*(z^{-1})=\Omega_{M'}(z)\Omega_{M'}^*(z^{-1})$. From (2.108), $\mathcal{W}(f)$ is given by

$$\mathcal{W}(f) = \frac{D(f)}{\sqrt{T}}\sum_i Y^*(f+\frac{i}{T})e^{-j2\pi(f+\frac{i}{T})t_o} \tag{2.119}$$

Using (2.118), the z-transform representing filter $\mathcal{W}(f)$ is

$$\Omega_{\mathcal{W}}(z) = z^{-H}\Omega_D(z)\Omega_Y^*(z^{-1}) \tag{2.120}$$

where

$$\Omega_D(z) = \sum_i d_i z^{-i} \tag{2.121}$$

$$\Omega_D(e^{j2\pi fT}) = D(f) \tag{2.122}$$

Depending on whether H is positive or negative, z^{-H} represents a delay or advance of $|H|T$ seconds. Replacing the z^{-H} term in (2.120) by unity simply means that at any particular sampling instant the receiver has detected symbol s_{i+H} instead of s_i. This presents no problem as long as the receiver knows which symbol it has detected. Thus the z^{-H} term can be neglected. Setting $H=0$ in (2.115) and (2.120) gives

$$r_k = \sum_{i=0}^{g} s_{k-i}y_i + n_k \tag{2.123}$$

$$\Omega_{\mathcal{W}}(z) = \Omega_D(z)\Omega_Y^*(z^{-1}) \tag{2.124}$$

The sampled impulse response seen at the output of the sampler, but before $\mathcal{W}(f)$, in fig. 2.5 has z-transform $\Omega_Y(z)$. Notice that for a near-MLSE, since $\Omega_D(z)=(\Omega_{M'}^*(z^{-1}))^{-1}$, $\Omega_{M'}(z)\Omega_{M'}^*(z^{-1})=\Omega_Y(z)\Omega_Y^*(z^{-1})$ and $\Omega_{M'}(z)$ has its roots on or inside the unit circle, then $\Omega_{\mathcal{W}}(z)$ has poles which are the roots of $\Omega_Y(z)$ lying outside the unit circle, and zeros which are the reciprocals of the complex conjugates of these roots.

2.6 Effect of Time-Varying Channel

The derivations so far in this chapter are, strictly speaking, based on the assumption that the transmission channel's impulse response is time-invariant. Basing the derivations on a general time-varying $h(t)$ serves to unecessarily complicate the analysis. This is because the rate of variation of the channel can, for all practical purposes, be assumed to be very much less than the baud rate, otherwise a decision-directed tracking device could never "learn" quickly enough about the changing medium. For HF channels the rate of variation is typically a few Hz or less, compared with a baud rate of 2400. Because of the time variation of $h(t)$, the bandwidth of $y(t)$ (the cascade of $a(t)$ and $h(t)$) is now not strictly limited to the bandwidth of $a(t)$, but since variations are considered very small c.f. T^{-1} we can reasonably assume that it is still approximately so. Thus our assumption in section 2.5 that $f_y \lesssim T^{-1}$, enabling us to define $\mathcal{B}(f)$ as a rectangular filter, can still be considered effectively valid for $a(t)$ bandlimited to around T^{-1} Hz.

We shall now re-examine the DFE and MFE, and include in the notation the time variation of the channel, and also the limitation of having a finite number of feedforward taps in the DFE.

2.6.1 MMSE DFE with Finite Number of Taps

For the MMSE DFE,

$$\Omega_D(z) = \frac{1}{m_0 \Omega_M^*(z^{-1})} \tag{2.125}$$

where $\Omega_M(z)$ was defined in (2.52). The coefficients $\{d_i\}$ for $i>0$ in (2.121) are zero, because $\Omega_M(z)$ has its roots on or inside the unit circle. Thus we can write

$$\Omega_{\mathcal{W}}(z) = \sum_{i=0}^{\infty} w_i z^i \tag{2.126}$$

The output of $\mathcal{W}(f)$ is

$$u_k = \sum_{i=0}^{\infty} r_{k+i} w_i = \sum_{i=-g}^{\infty} s_{k+i} \sum_{h=0}^{g} y_h w_{h+i} + \sum_{i=0}^{\infty} n_{k+i} w_i \tag{2.127}$$

It can be seen that in detecting s_k from u_k, the DFE need only subtract the ISI contributions from the previous g symbols. Eq. (2.126) assumes that $\Omega_{\mathcal{W}}(z)$ consists of an infinite number of terms, so that a digital filter implementation of $\mathcal{W}(f)$ would consist of an infinite number of taps. In practice this cannot be so. We shall restrict the number of taps to N. Fig. 2.6 shows the form of the DFE, consisting of an N-tap linear *feedforward* section and a g-tap *feedback* section that subtracts the ISI from previous symbols, assuming past decisions are correct. This DFE will in future be referred to as DFE(N,g). Unfortunately the optimum MMSE DFE tap-weights change with the value of N, and so it will be necessary to re-do the minimization

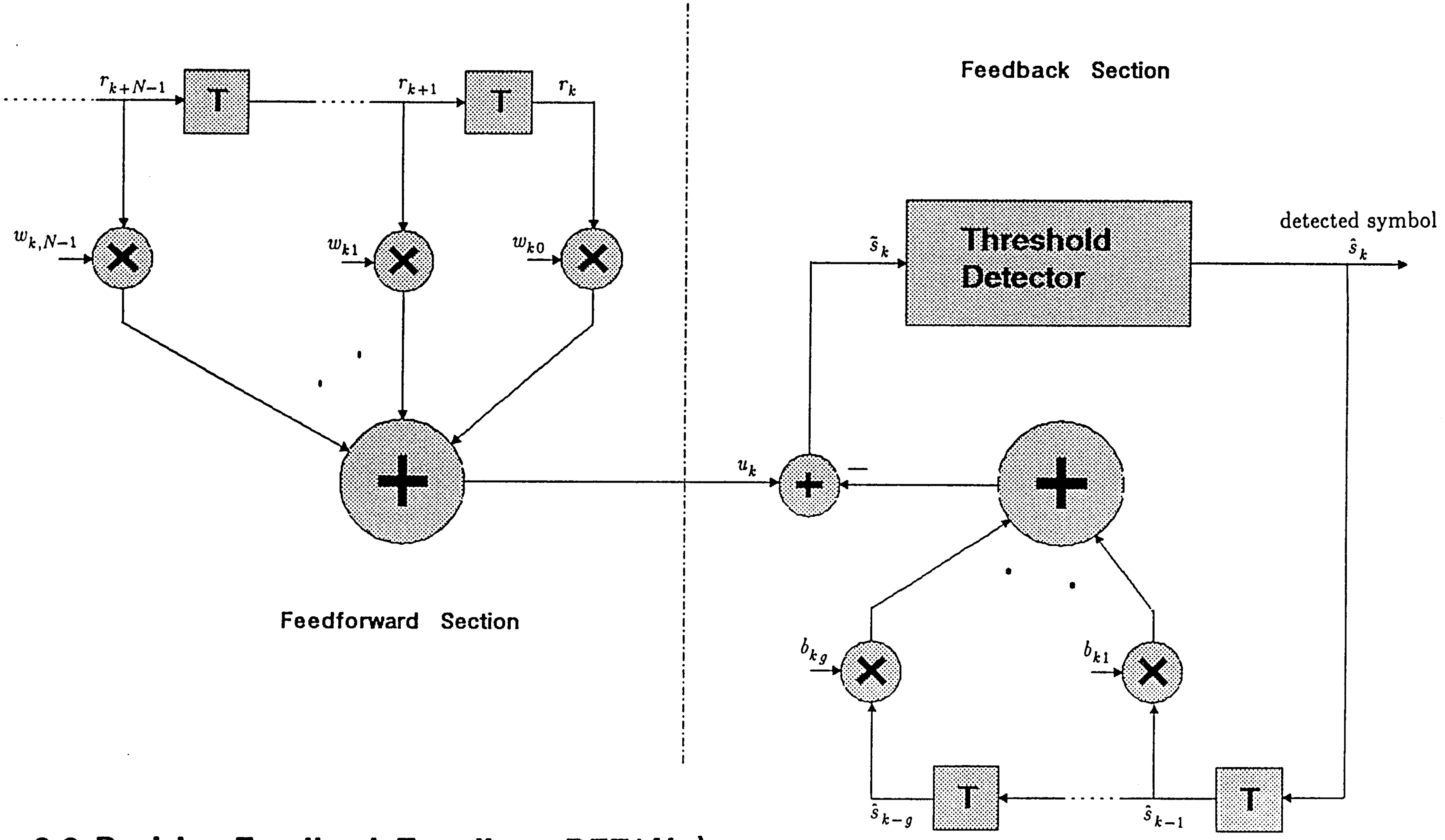


Fig. 2.6 Decision Feedback Equalizer: DFE(*N,g*)

process for their determination.

The possibility that *Y(f)* can vary with time has been mentioned, but up till now has not been explicitly represented in the mathematics. Let us characterize the sampled impulse response $y_0, \ldots, y_g$ in (2.123) by a subscript k to denote its time dependence. Thus

$$r_k = \sum_{i=0}^{g} s_{k-i} y_{ki} + n_k \tag{2.128}$$

Define the following column vectors, where superscript "t" denotes the transpose, as

$$\underline{\mathbf{y}}_{ki}^{\mathrm{t}} = [y_{k,0-i} \cdots y_{k+p,p-i} \cdots y_{k+N-1,N-1-i}] \tag{2.129}$$

$$\underline{\mathbf{w}}_{k}^{\mathrm{t}} = [w_{k0} \cdots w_{k,N-1}] \tag{2.130}$$

$$\underline{\mathbf{n}}_{k}^{\mathrm{t}} = [n_k \ldots n_{k+N-1}] \tag{2.131}$$

Note that $\underline{\mathbf{w}}_k$ also has a subscript to denote time dependence. Assuming past decisions are correct, the error in $\tilde{s}_k$ is (see fig. 2.6)

$$e_k = \tilde{s}_k - s_k = \underline{\mathbf{w}}_k^{\mathrm{t}}[\sum_{i=-g}^{N-1} s_{k+i}\underline{\mathbf{y}}_{ki} + \underline{\mathbf{n}}_k] - \sum_{i=1}^{g} s_{k-i} b_{ki} - s_k \tag{2.132}$$

We know that in order to minimize the MSE,

$$b_{ki} = \underline{\mathbf{w}}_k^{\mathrm{t}} \underline{\mathbf{y}}_{k,-i} \qquad \text{for} \quad 1 \leq i \leq g \tag{2.133}$$

Substituting (2.133) into (2.132) and denoting the MSE as ξ_k we get

$$\xi_k = \mathrm{E}[|e_k|^2] = \sigma_s^2 \Big\{ \underline{\mathbf{w}}_k^{\mathrm{t}} [\sum_{i=0}^{N-1} \underline{\mathbf{y}}_{ki} \underline{\mathbf{y}}_{ki}^{*\mathrm{t}} + \rho \mathrm{I}] \underline{\mathbf{w}}_k^{*} - \underline{\mathbf{w}}_k^{\mathrm{t}} \underline{\mathbf{y}}_{k0} - (\underline{\mathbf{w}}_k^{\mathrm{t}} \underline{\mathbf{y}}_{k0})^{*} + 1 \Big\} \tag{2.134}$$

I is the $N \times N$ identity matrix. Let us define the $N \times N$ matrix Y_k as

$$\mathrm{Y}_k = \sum_{i=0}^{N-1} \underline{\mathbf{y}}_{ki} \underline{\mathbf{y}}_{ki}^{*\mathrm{t}} + \rho \mathrm{I} \tag{2.135}$$

Note that Y_k is Hermetian, i.e. $\mathrm{Y}_k = \mathrm{Y}_k^{*\mathrm{t}}$. Differentiating ξ_k with respect to $\underline{\mathbf{w}}_k$ we obtain

$$\delta\xi_k = 2\mathrm{Re}[\sigma_s^2 \{ \underline{\mathbf{w}}_k^{\mathrm{t}} \mathrm{Y}_k - \underline{\mathbf{y}}_{k0}^{*\mathrm{t}} \} \delta \underline{\mathbf{w}}_k^{*}] \tag{2.136}$$

Hence $\delta\xi_k=0$ is given by

$$\underline{\mathbf{w}}_k = \mathbf{Y}_k^{*-1}\underline{\mathbf{y}}_{k0}^* \tag{2.137}$$

where superscript "-1" denotes the inverse of a matrix. Substituting (2.137) into (2.134) yields

$$\xi_{k,\min} = \sigma_s^2(1-\underline{\mathbf{w}}_k^{\mathrm{t}}\mathbf{Y}_k\underline{\mathbf{w}}_k^*) = \sigma_s^2(1-\underline{\mathbf{y}}_{k0}^{*\mathrm{t}}\mathbf{Y}_k{}^{-1}\underline{\mathbf{y}}_{k0}) = \sigma_s^2(1-\underline{\mathbf{w}}_k^{\mathrm{t}}\underline{\mathbf{y}}_{k0}) \tag{2.138}$$

We can express $\xi_{k,\min}$ in another useful form by first substituting for $\mathbf{Y}_k$ in $\mathbf{Y}_k^*\underline{\mathbf{w}}_k=\underline{\mathbf{y}}_{k0}^*$ and then comparing the first coefficient to get

$$y_{k0}^*\underline{\mathbf{w}}_k^{\mathrm{t}}\underline{\mathbf{y}}_{k0}+w_{k0}\rho = y_{k0}^* \tag{2.139}$$

Thus

$$\xi_{k,\min} = \sigma_s^2\rho\frac{w_{k0}}{y_{k0}^*} = \sigma_n^2\frac{w_{k0}}{y_{k0}^*} \tag{2.140}$$

As N increases, the tap weight vector $\underline{\mathbf{w}}_k$ approaches the first N taps in $\mathcal{W}(f)$ for the MMSE DFE(∞,g) solution. Thus if N is large enough, and ρ is small, $\underline{\mathbf{w}}_k$ can be used as an approximation to the first N taps in $\mathcal{W}(f)$ for the near-MLSE, the feedback section of the MMSE DFE(N,g) plus $b_{k0}=1$ representing the ($g+1$)-length sampled impulse response seen by the near-MLSE.

2.6.2 MMSE MFE and ZF MFE

For the MMSE MFE,

$$\Omega_D(z) = \frac{1}{\rho+E_n} \tag{2.141}$$

Then from (2.124),

$$\Omega_{\mathcal{W}}(z) = \frac{\Omega_Y^*(z^{-1})}{(\rho+E_n)} \tag{2.142}$$

$\Omega_D(z)$ and $\Omega_{\mathcal{W}}(z)$ for the ZF MFE can be obtained from (2.141) and (2.142) by setting $\rho=0$. $\Omega_{\mathcal{W}}(z)$ can be written as

$$\Omega_{\mathcal{W}}(z) = \sum_{i=0}^{g} w_i z^i \tag{2.143}$$

The MFE is depicted in fig. 2.7, and will sometimes be referred to as MFE($g+1$) because it has $g+1$ taps in the digital filter $\mathcal{W}(f)$. Using equations (2.128)-(2.131) but with N replaced by $g+1$, and assuming correct decisions for past and future symbols, we get for the error in $\tilde{s}_k$

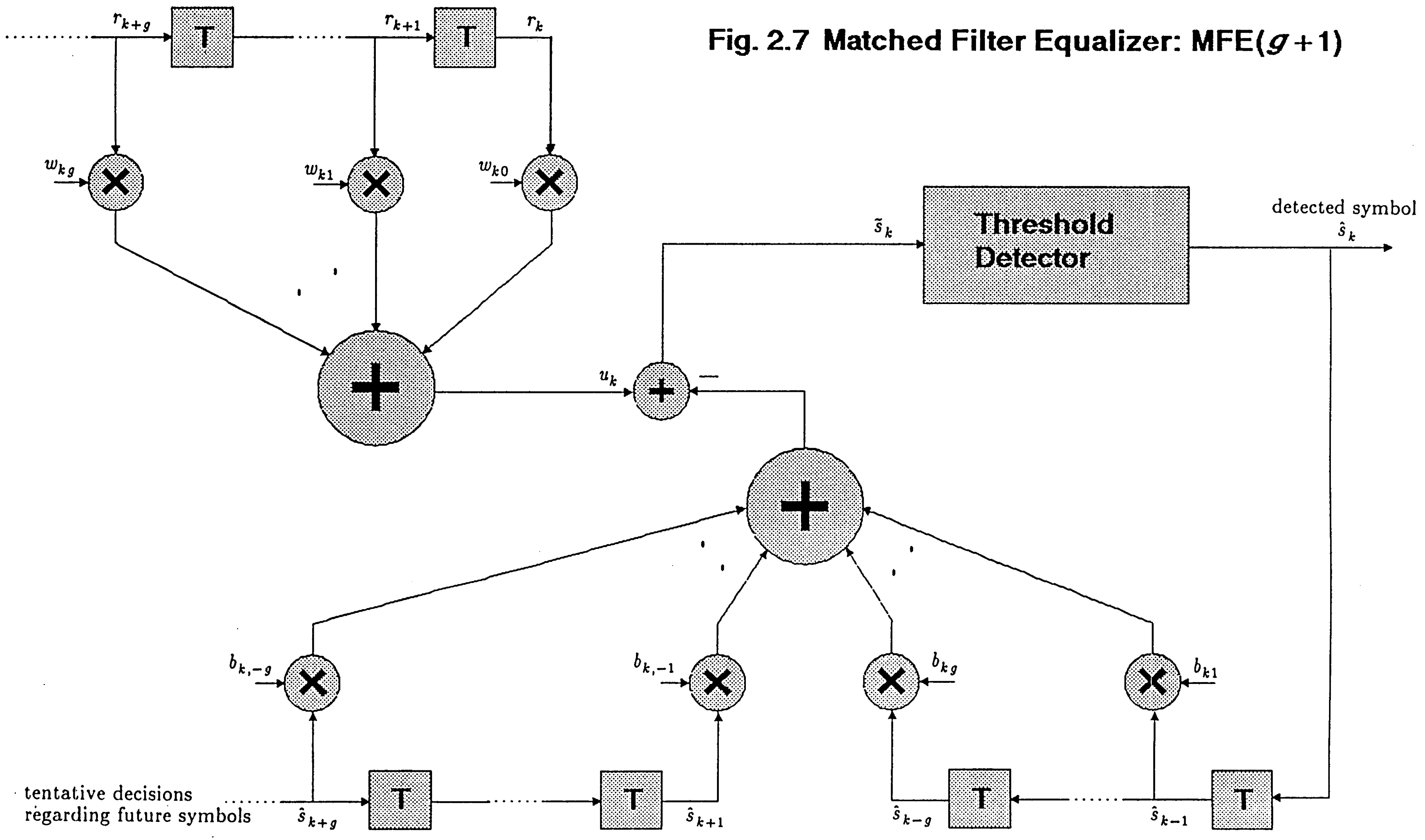


Fig. 2.7 Matched Filter Equalizer: MFE($g+1$)

$$e_k = \tilde{s}_k - s_k = \underline{\mathbf{w}}_k^{\mathrm{t}}[\sum_{i=-g}^{g} s_{k+i}\underline{\mathbf{y}}_{ki} + \underline{\mathbf{n}}_k] - \sum_{i=-g}^{g} s_{k-i} b_{ki} - s_k \tag{2.144}$$

For the MMSE or ZF criterion we need

$$b_{ki} = \underline{\mathbf{w}}_k^{\mathrm{t}}\underline{\mathbf{y}}_{k,-i} \qquad -g \leq i \leq -1,\ 1 \leq i \leq g \tag{2.145}$$

so that

$$e_k = s_k(\underline{\mathbf{w}}_k^{\mathrm{t}}\underline{\mathbf{y}}_{k0} - 1) + \underline{\mathbf{w}}_k^{\mathrm{t}}\underline{\mathbf{n}}_k \tag{2.146}$$

For the MMSE criterion it is straightforward to show that

$$\underline{\mathbf{w}}_k = \frac{\underline{\mathbf{y}}_{k0}^{*}}{(\rho + \|\underline{\mathbf{y}}_{k0}\|^2)} \qquad \text{(MMSE)} \tag{2.147}$$

where $\|\underline{\mathbf{y}}_{k0}\|^2 = \underline{\mathbf{y}}_{k0}^{*\mathrm{t}}\underline{\mathbf{y}}_{k0}$ is the magnitude (or norm) of the vector $\underline{\mathbf{y}}_{k0}$. Setting $\rho=0$ in (2.147) gives us the result for the ZF criterion

$$\underline{\mathbf{w}}_k = \frac{\underline{\mathbf{y}}_{k0}^{*}}{\|\underline{\mathbf{y}}_{k0}\|^2} \qquad \text{(ZF)} \tag{2.148}$$

The final MSE takes the form

$$\xi_k = \frac{\sigma_s^2 \rho}{(\rho + \|\underline{\mathbf{y}}_{k0}\|^2)} \qquad \text{(MMSE)} \tag{2.149}$$

$$\xi_k = \frac{\sigma_s^2 \rho}{\|\underline{\mathbf{y}}_{k0}\|^2} \qquad \text{(ZF)} \tag{2.150}$$

Notice that in introducing a time subscript k, $\Omega_Y(z)$ and E_n in (2.142) have been replaced by $\sum_{i=0}^{g} y_{k+i,i} z^{-i}$ and $\|\underline{\mathbf{y}}_{k0}\|^2$ respectively.

2.7 Other Detection Devices

We will briefly mention a few less commonly used detection devices.

The probabilistic symbol-by-symbol equalization algorithm [3] is based on computation of a-posteriori probabilities, and is optimum in the sense of minimizing the probability of symbol error, in contrast to the MLSE which minimizes the probability of sequence error. There is also a delay in detection. It requires knowledge of the channel sampled impulse response, and involves a large number of computations per received sample. In particular, it involves summations of exponential factors. Therefore the major shortcoming of this device is its heavy computational burden. Monte Carlo simulations have shown it to be superior in performance to the DFE, and comparable to the MLSE [3].

The linear feedback equalizer [3] is similar to the DFE except that the feedback filter contains the pre-threshold estimates $\{\tilde{s}_k\}$. Drawbacks with this device are that it is only marginally superior to the linear equalizer, and adaptive versions of it are prone to instability [3].

Another device is the so-called message estimator [56], which is based on the Kalman filter. It requires knowledge of the channel sampled impulse reponse, and in an adaptive implementation it has a faster start-up convergence than do the conventional adaptive implementations of the LE and DFE. However, simulations have shown that it is only marginally superior in performance to the LE, and significantly inferior to the DFE [56].

Chapter 3
TRACKING ALGORITHMS

We shall now describe two algorithms that are often used to provide adaptation of receiver filters under time-varying channel conditions. Let us first define the following.

$$\underline{x}_k^{\mathrm{t}} = [x_{k1}\ x_{k2} \cdots x_{kM}] \tag{3.1}$$

$$\underline{c}_i^{\mathrm{t}} = [c_{i1}\ c_{i2} \cdots c_{iM}] \tag{3.2}$$

$$\epsilon_{ki} = z_k - \underline{x}_k^{\mathrm{t}} \underline{c}_i \tag{3.3}$$

The M-length vectors $\underline{x}_k$ and $\underline{c}_i$ are termed the input vector and estimated vector respectively. The quantities z_k and ϵ_{ki}, both scalars, are termed the desired value and error respectively. In the context of adaptive equalization, $\underline{x}_k$ and z_k can, in general, be regarded as linear functions of the transmitted data symbols $\{s_i\}$, the noise samples $\{n_i\}$ and the channel sampled impulse response. Assume that at time kT we know $\underline{x}_i$ and z_i for $i \leq k$.

3.1 The Steepest Descent (SD) Algorithm

This is also known as the MSE algorithm [3], or the least-mean-squares (LMS) algorithm [24]. At time kT we wish to determine $\underline{c}_i$ such that the mean square error $\xi_k = \mathrm{E}[|\epsilon_{ki}|^2]$ is minimized, the expectation being over the data symbol and noise sequences.

$$\delta\xi_k = -2\mathrm{Re}\{\delta\underline{c}_i^{*\mathrm{t}} \mathrm{E}[\underline{x}_k^* \epsilon_{ki}]\} = 2\mathrm{Re}\{\delta\underline{c}_i^{*\mathrm{t}}(\mathrm{E}[\underline{x}_k^* \underline{x}_k^{\mathrm{t}}]\underline{c}_i - \mathrm{E}[\underline{x}_k^* z_k])\} \tag{3.4}$$

Clearly, ξ_k is minimized when

$$\underline{c}_i = \underline{c}_{k,\mathrm{opt}} = (\mathrm{E}[\underline{x}_k^* \underline{x}_k^{\mathrm{t}}])^{-1} \mathrm{E}[\underline{x}_k^* z_k] \tag{3.5}$$

ξ_k can be viewed as a quadratic surface in M-dimensional space, a particular vector $\underline{c}_i$ corresponding to a particular point on that surface. We could find $\underline{c}_{k,\mathrm{opt}}$ iteratively by observing the gradient $\partial\xi_k/\partial\underline{c}_i$ and adjusting $\underline{c}_i$ by a small amount in the direction of steepest descent. From (3.4) it can be seen that the direction of steepest descent is the same as that of vector $\mathrm{E}[\underline{\mathbf{x}}_k^* \epsilon_{ki}]$. An updated vector $\underline{\mathbf{c}}_{i+1}$ is thus given by

$$\underline{\mathbf{c}}_{i+1} = \underline{\mathbf{c}}_i + \mu \mathrm{E}[\underline{\mathbf{x}}_k^* \epsilon_{ki}] \tag{3.6}$$

where μ is a small positive real number. Subtracting $\underline{\mathbf{c}}_{k,\mathrm{opt}}$ from both sides of (3.6) we get

$$\underline{\mathbf{c}}_{i+1} - \underline{\mathbf{c}}_{k,\mathrm{opt}} = (\mathbf{I} - \mu \mathbf{R}_k)(\underline{\mathbf{c}}_i - \underline{\mathbf{c}}_{k,\mathrm{opt}}) \tag{3.7}$$

where

$$\mathbf{R}_k = \mathrm{E}[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^{\mathrm{t}}] = \mathbf{V}_k \mathbf{\Lambda}_k \mathbf{V}_k^{*\mathrm{t}} \tag{3.8}$$

$\mathbf{V}_k$ is the matrix of eigenvectors and $\mathbf{\Lambda}_k$ the diagonal matrix of positive real eigenvalues $\{\lambda_{ki}\}$, $1 \leq i \leq M$, of the covariance matrix $\mathbf{R}_k$ ($\mathbf{V}_k \mathbf{V}_k^{*\mathrm{t}} = \mathbf{I}$). Define the error vector $\underline{\mathbf{d}}_{ki}$ as

$$\underline{\mathbf{d}}_{ki} = \underline{\mathbf{c}}_i - \underline{\mathbf{c}}_{k,\mathrm{opt}} \tag{3.9}$$

Then from (3.7),

$$\underline{\mathbf{d}}_{k,i+1} = (\mathbf{I} - \mu \mathbf{R}_k) \underline{\mathbf{d}}_{ki} \tag{3.10}$$

From (3.10) we can see that $\|\underline{\mathbf{d}}_{ki}\|^2 = \underline{\mathbf{d}}_{ki}^{*\mathrm{t}} \underline{\mathbf{d}}_{ki}$ converges to a steady-state value (0 in this case) as $i \rightarrow \infty$ if

$$\|\underline{\mathbf{d}}_{k,i+1}\|^2 < \|\underline{\mathbf{d}}_{ki}\|^2 \tag{3.11}$$

where

$$\|\underline{\mathbf{d}}_{k,i+1}\|^2 = \underline{\mathbf{d}}_{ki}^{*\mathrm{t}} \mathbf{V}_k (\mathbf{I} - \mu \mathbf{\Lambda}_k)^2 \mathbf{V}_k^{*\mathrm{t}} \underline{\mathbf{d}}_{ki} \tag{3.12}$$

Eq. (3.11) is thus satisfied if

$$(1 - \mu \lambda_{ki})^2 < 1 \qquad \text{for } 1 \leq i \leq M \tag{3.13}$$

$$\Rightarrow \qquad 0 < \mu < \frac{2}{\lambda_{k,\mathrm{max}}} \tag{3.14}$$

where $\lambda_{k,\mathrm{max}}$ is the largest eigenvalue of $\mathbf{R}_k$. In practice the gradient quantity $\mathrm{E}[\underline{\mathbf{x}}_k^* \epsilon_{ki}]$ is unknown, so it is replaced in (3.6) by the unbiased estimate $\underline{\mathbf{x}}_k^* \epsilon_{ki}$. Further, if we carry out the iterative process recursively in time, (3.6) becomes

$$\underline{\mathbf{c}}_k = \underline{\mathbf{c}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \epsilon_k \tag{3.15}$$

where

$$\epsilon_k = \epsilon_{k,k-1} = z_k - \underline{\mathbf{x}}_k^{\mathrm{t}} \underline{\mathbf{c}}_{k-1} \tag{3.16}$$

Eq. (3.15) is the basic form of the SD algorithm that is commonly used for adaptation purposes. The vector $\underline{\mathbf{c}}_{k-1}$ is computed at time $(k-1)T$ and is used at time kT as an estimate of $\underline{\mathbf{c}}_{k,\text{opt}}$. The update $\underline{\mathbf{c}}_k$ is an improved estimate of $\underline{\mathbf{c}}_{k,\text{opt}}$. Let the error vector $\underline{\mathbf{d}}_k$ be given by

$$\underline{\mathbf{d}}_k = \underline{\mathbf{c}}_k - \underline{\mathbf{c}}_{k+1,\text{opt}} \tag{3.17}$$

Then from (3.15),

$$\underline{\mathbf{d}}_k = (\mathbf{I} - \mu \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^{\mathbf{t}}) \underline{\mathbf{d}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \epsilon_{k,\text{opt}} - (\underline{\mathbf{c}}_{k+1,\text{opt}} - \underline{\mathbf{c}}_{k,\text{opt}}) \tag{3.18}$$

where

$$\epsilon_{k,\text{opt}} = z_k - \underline{\mathbf{x}}_k^{\mathbf{t}} \underline{\mathbf{c}}_{k,\text{opt}} \tag{3.19}$$

At time kT, for $\mathrm{E}[\, \|\underline{\mathbf{d}}_k\|^2]$ to be able to reach a steady-state condition it is required that the first term on the RHS of (3.18) be such that

$$\mathrm{E}[\, \|(\mathbf{I} - \mu \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^{\mathbf{t}}) \underline{\mathbf{d}}_{k-1}\|^2] < \mathrm{E}[\, \|\underline{\mathbf{d}}_{k-1}\|^2] \tag{3.20}$$

If we make the common assumption [24] that the input vectors are statistically independent, i.e.

$$\mathrm{E}[\underline{\mathbf{x}}_i^* \underline{\mathbf{x}}_m^{\mathbf{t}}] = 0 \qquad \text{for} \quad i \neq m \tag{3.21}$$

then, since $\underline{\mathbf{d}}_{k-1}$ depends on $\underline{\mathbf{x}}_i$ for $i \leq k-1$, (3.20) becomes

$$\mathrm{E}[\underline{\mathbf{d}}_{k-1}^{*\mathbf{t}} (\mathbf{I} - 2\mu \mathrm{R}_k + \mu^2 \sum_{i=1}^{M} \lambda_{ki} \mathrm{R}_k) \underline{\mathbf{d}}_{k-1}] < \mathrm{E}[\, \|\underline{\mathbf{d}}_{k-1}\|^2] \tag{3.22}$$

where it has been assumed that

$$\mathrm{E}[(\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^{\mathbf{t}})^2] \approx \mathrm{E}[\underline{\mathbf{x}}_k^{\mathbf{t}} \underline{\mathbf{x}}_k^*] \mathrm{R}_k = \sum_{i=1}^{M} \lambda_{ki} \mathrm{R}_k \tag{3.23}$$

Eq. (3.22) is satisfied if

$$(1 - 2\mu \lambda_{km} + \mu^2 \sum_{i=1}^{M} \lambda_{ki} \lambda_{km}) < 1 \qquad \text{for} \quad 1 \leq m \leq M \tag{3.24}$$

$$\Rightarrow \qquad 0 < \mu < \frac{2}{\sum_{i=1}^{M} \lambda_{ki}} \tag{3.25}$$

Contrast the permissible range for μ in (3.25) with that for the deterministic case (i.e. $E[\underline{x}_k^*\epsilon_k]$ known) in (3.14). With equal eigenvalues, the range in (3.25) is reduced c.f. (3.14) by the factor *M*. A more detailed derivation of (3.25) can be found in [57], where it first appeared. Note that the steady-state value of $E[\,\|\underline{d}_k\|^2]$ varies with time integer k if $\underline{c}_{k,\mathrm{opt}}$ does. Even for $\underline{c}_{k,\mathrm{opt}}$ invariant with respect to k, $E[\,\|\underline{d}_k\|^2]$ is never going to reach zero if $E[|\epsilon_{k,\mathrm{opt}}|^2]$ is not zero [3], [57], because then we will always be using a "noisy" estimate $\underline{x}_k^*\epsilon_k$ of the gradient term $E[\underline{x}_k^*\epsilon_k]$.

It is shown in [57], for a time-invariant $\underline{c}_{k,\mathrm{opt}}$, that the speed of convergence of the algorithm from any initial starting point of the estimated vector $\underline{c}_k$ is dependent on the vector length M and, to a lesser degree, on the eigenvalue spread of $\mathbf{R}_k$, characterized by the ratio $\lambda_{k,\mathrm{max}}/\lambda_{k,\mathrm{min}}$. A larger M and/or $\lambda_{k,\mathrm{max}}/\lambda_{k,\mathrm{min}}$ produces slower convergence, which therefore means that the algorithm is less able to track time variations in $\underline{c}_{k,\mathrm{opt}}$. To remove the effect, should the need be, of eigenvalue spread, consider the transformed input vector $\underline{x}'_k$ given by

$$\underline{x}'_k = \Lambda_k^{-1/2}V_k^t\underline{x}_k \tag{3.26}$$

and the estimated vector $\underline{c}'_k$ given recursively as

$$\underline{c}'_k = \underline{c}'_{k-1}+\mu\underline{x}'^{*}_k\epsilon'_k \tag{3.27}$$

where

$$\epsilon'_k = z_k-\underline{x}'^{t}_k\underline{c}'_{k-1} \tag{3.28}$$

The eigenvaules of $E[\underline{x}'^{*}_k\underline{x}'^{t}_k]=I$ are all equal to unity, so that the spread is minimized. The optimum value $\underline{c}_{k,\mathrm{opt}}$ (3.5) is related to $\underline{c}'_{k,\mathrm{opt}}$ by

$$E[\underline{x}'^{*}_k z_k] = \underline{c}'_{k,\mathrm{opt}} = \Lambda_k^{1/2}V_k^{*t}\underline{c}_{k,\mathrm{opt}} \tag{3.29}$$

Thus if we multiply $\underline{c}'_k$ by $V_k\Lambda_k^{-1/2}$ we get an approximation to $\underline{c}_{k,\mathrm{opt}}$. Denoting this, as before, by $\underline{c}_k$ and assuming

$$V_k\Lambda_k^{-1/2}\underline{c}'_{k-1} \approx V_{k-1}\Lambda_{k-1}^{-1/2}\underline{c}'_{k-1} = \underline{c}_{k-1} \tag{3.30}$$

then if we multiply both sides of (3.27) by $\mathbf{V}_k\mathbf{\Lambda}_k^{-1/2}$ we get the algorithm

$$\underline{c}_k = \underline{c}_{k-1} + \mu \mathbf{R}_k^{-1}\underline{x}_k\epsilon_k \tag{3.31}$$

with ϵ_k given again by (3.16). This algorithm is called an *orthogonal* SD algorithm [24], because the input vector $\underline{x}_k'$ has uncorrelated elements. Its speed of convergence is independent of the eigenvalue spread in $\mathbf{R}_k$, and so it will be much faster than the algorithm of (3.15) for a large $\lambda_{k,\max}/\lambda_{k,\min}$ ratio. The problem with its implementation, of course, lies in having knowledge of $\mathbf{R}_k$. The range of μ for stable convergence is

$$0 < \mu < \frac{2}{M} \tag{3.32}$$

Further details on the SD algorithm can be found in [3], [24], [57]-[60]. The SD algorithm of (3.15) is just about the simplest adaptive process around. At each iteration it requires $10M$ equivalent real multiplications and $8M$ real additions.

3.2 Recursive Least Squares (RLS) Algorithms

In the SD algorithm we strive to minimize a statistical average of the squared error $|\epsilon_k|^2$. In the least squares approach we strive to minimize a time average of the squared error, as defined below.

$$\bar{\xi}_k = \sum_{i=0}^{k}\omega^{k-i}|\epsilon_{ik}|^2 = \sum_{i=0}^{k}\omega^{k-i}|z_i - \underline{x}_i^t\underline{c}_k|^2 \tag{3.33}$$

where ω is a weighting coefficient, $0<\omega\leq 1$. Eq. (3.33) assumes an average over $k+1$ error values, the weighting for each value getting exponentially smaller as we go "into the past". We wish to determine $\underline{c}_k$ such that $\bar{\xi}_k$ is minimized.

$$\delta\bar{\xi}_k = 2\mathrm{Re}\{\delta\underline{c}_k^{*t}[\sum_{i=0}^{k}\omega^{k-i}(\underline{x}_i^*\underline{x}_i^t\underline{c}_k - \underline{x}_i^* z_i)]\} \tag{3.34}$$

Thus for minimum $\bar{\xi}_k$,

$$\underline{c}_k = (\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^*\underline{x}_i^t)^{-1}(\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^* z_i) \tag{3.35}$$

Notice the similarity between the solution here and that of (3.5); the statistical averages have been replaced by weighted time averages. With a little effort it is possible to show, for $k>0$, that

$$\underline{c}_k = \underline{c}_{k-1}+(\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^*\underline{x}_i^{t})^{-1}\underline{x}_k^*(z_k-\underline{x}_k^{t}\underline{c}_{k-1}) \tag{3.36}$$

We can derive a recursive formula for the evaluation of the inverted matrix in (3.36) as follows. Suppose we have an $M\times M$ matrix P_k which is related to P_{k-1} by

$$\mathrm{P}_k^{-1} = \omega\mathrm{P}_{k-1}^{-1} + \tfrac{\omega}{\eta}\underline{x}_k^*\underline{x}_k^{t} \tag{3.37}$$

where η is a positive real constant. Then

$$\mathrm{P}_k^{-1} = \omega^{k+1}\mathrm{P}_{-1}^{-1} + \tfrac{\omega}{\eta}(\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^*\underline{x}_i^{t}) \tag{3.38}$$

As $\omega\leq 1$, for k large enough we can neglect the first term in comparison to the second on the RHS of (3.38). Thus

$$\mathrm{P}_k^{-1} \approx \tfrac{\omega}{\eta}(\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^*\underline{x}_i^{t}) \qquad \text{for } k \text{ large} \tag{3.39}$$

Note that P_{-1}^{-1} cannot be a zero matrix because then P_0^{-1} has no inverse. It is usual, therefore, to set P_{-1} to the identity matrix, which then ensures that the first few iterations of P_k^{-1} are not ill-conditioned. Ill-conditioning of P_k^{-1} means that small deviations in the elements of the matrix can lead to large changes in the inverse P_k. Large errors and instability can then arise because of computer round-off errors, the degree depending on the numerical precision available.

Consider now the recursive formula

$$\underline{c}_k = \underline{c}_{k-1}+\tfrac{\omega}{\eta}\mathrm{P}_k\underline{x}_k^*(z_k-\underline{x}_k^{t}\underline{c}_{k-1}) \tag{3.40}$$

If at some high enough k, P_k is approximated by (3.39) and $\underline{c}_{k-1}$ is the least squares solution at time $(k-1)T$, then $\underline{c}_k$ will, from (3.36), be the least squares solution at time kT. We need to show that from any initial starting vector $\underline{c}_{-1}$, $\underline{c}_k$ does converge to the least squares solution. With a little rearranging of (3.40) and substituting for $\underline{x}_k^*\underline{x}_k^{t}$ from (3.37), it is possible to show that

$$\mathrm{P}_k^{-1}\underline{\mathbf{c}}_k = \omega\mathrm{P}_{k-1}^{-1}\,\underline{\mathbf{c}}_{k-1}+\tfrac{\omega}{\eta}\underline{\mathbf{x}}_k^{*}z_k = \omega^{k+1}\mathrm{P}_{-1}^{-1}\,\underline{\mathbf{c}}_{-1}+\tfrac{\omega}{\eta}\sum_{i=0}^{k}\omega^{k-i}\underline{\mathbf{x}}_i^{*}z_i \tag{3.41}$$

For P_{-1} and $\underline{\mathbf{c}}_{-1}$ fixed at some initial value, the first term on the RHS of (3.41) becomes negligible in comparison to the second for k large enough, and thus $\underline{\mathbf{c}}_k$ tends to the solution in (3.35). It is usual to set $\underline{\mathbf{c}}_{-1}=\underline{\mathbf{0}}$, so the first term in (3.41) is zero anyway. The algorithm can now be stated. At time kT the necessary ordered computations for updating $\underline{\mathbf{c}}_{k-1}$ to $\underline{\mathbf{c}}_k$ are:

$$\left.\begin{array}{lll}
(i) & \epsilon_k = z_k-\underline{\mathbf{x}}_k^{\mathrm{t}}\underline{\mathbf{c}}_{k-1} & \\
(ii) & \underline{\mathbf{k}}_k = (\eta+\underline{\mathbf{x}}_k^{\mathrm{t}}\mathrm{P}_{k-1}\underline{\mathbf{x}}_k^{*})^{-1}\mathrm{P}_{k-1}\underline{\mathbf{x}}_k^{*} & (=\tfrac{\omega}{\eta}\mathrm{P}_k\underline{\mathbf{x}}_k^{*}) \\
(iii) & \mathrm{P}_k = \tfrac{1}{\omega}(\mathrm{P}_{k-1}-\underline{\mathbf{k}}_k\underline{\mathbf{x}}_k^{\mathrm{t}}\mathrm{P}_{k-1}) & \\
(iv) & \underline{\mathbf{c}}_k = \underline{\mathbf{c}}_{k-1}+\underline{\mathbf{k}}_k\epsilon_k &
\end{array}\right]\text{ (i)–(iv)} \tag{3.42}$$

The parameter η controls the initial convergence of the algorithm from start. A large η means that we need to wait longer before (3.39) becomes valid, and too small a η means that the first few iterations of P_k^{-1} will become ill-conditioned. The weighting coefficient ω also influences the initial convergence, but is primarily viewed as the parameter that controls adaptability to time-varying conditions. The faster the change of $\underline{\mathbf{c}}_{k,\mathrm{opt}}$ (3.5) with k, the smaller must ω be in order to keep track. However, the smaller the value of ω the greater the influence of noise, and too small a ω also leads to ill-conditioning of P_k^{-1}. For $\underline{\mathbf{c}}_{k,\mathrm{opt}}$ time-invariant, ω can be set close to unity.

The RLS algorithm described here is often referred to as the RLS Kalman (RLSK), because it is a variant of the Kalman filter algorithm [61]. It is recognised as the fastest known adaptive process [24], [62]. The vector $\underline{\mathbf{k}}_k$ in (3.42) is frequently called the Kalman gain vector. The use of the Kalman filter algorithm for adaptive equalization was first demonstrated in [61]. For a time-invariant $\underline{\mathbf{c}}_{k,\mathrm{opt}}$, it is demonstrated in [61] that the speed of convergence (with $\omega=1$) is fairly insensitive to η, provided η is reasonably small but not zero (we will demonstrate its influence later on in chapters 6 and 7). Further, the convergence time can be shown to be of the order of M iterations, which is usually about 3–10 times faster than the SD algorithm [63].

Comparison of the RLSK (3.40) with the orthogonal SD (3.31) suggests that the high speed is achieved because we are performing a kind of self-orthogonalization with the matrix $\mathbf{P}_k$, so that insensitivity to the eigenvalue spread of $\mathbf{R}_k$ is obtained. This is true, in that the RLSK algorithm is insensitive to eigenvalue spread [3], but still its convergence can be considerably faster than the orthogonal SD algorithm [63]. As to why this might be so, consider the situation in which we have the M input vectors and desired values $\underline{\mathbf{x}}_i$, z_i, $0 \leq i \leq M-1$. Then, assuming $\underline{\mathbf{c}}_{i,\mathrm{opt}}$ does not vary much over $0 \leq i \leq M-1$, to determine the M elements of $\underline{\mathbf{c}}_{M-1}$ we have the M unique equations (assuming the $\{\underline{\mathbf{x}}_i\}$ are linearly independent of each other)

$$\underline{\mathbf{x}}_i^{\mathrm{t}} \underline{\mathbf{c}}_{M-1} = z_i \qquad 0 \leq i \leq M-1 \tag{3.43}$$

Weighting the i^{th} equation with ω^{M-1-i} and multiplying both sides by $\underline{\mathbf{x}}_i^*$ we can solve (3.43) to get

$$\underline{\mathbf{c}}_{M-1} = (\sum_{i=0}^{M-1} \omega^{M-1-i} \underline{\mathbf{x}}_i^* \underline{\mathbf{x}}_i^{\mathrm{t}})^{-1} (\sum_{i=0}^{M-1} \omega^{M-1-i} \underline{\mathbf{x}}_i^* z_i) \tag{3.44}$$

Thus, if the influence of P_{-1} in the RLSK algorithm is negligible by about M iterations we will have obtained an estimated vector that is quite close to its steady-state value. A more detailed explanation of the fast convergence of RLS algorithms can be found in [63].

A problem with the RLSK algorithm is that it can become numerically unstable [3], [16], [64]. The main reason for this is that $\mathbf{P}_k$ is computed as the difference of two positive semi-definite matrices. Computer round-off errors mean that with each iteration the numerical accuracy in $\mathbf{P}_k$ is reduced. This may result in a $\mathbf{P}_k$ matrix which is indefinite, i.e. having both positive and negative eigenvalues. To alleviate the problem, algorithms have been developed which avoid the computation of $\mathbf{P}_k$ according to (3.42). These are based on the decomposition of $\mathbf{P}_k$ in the form

$$\mathbf{P}_k = \mathbf{U}_k^* \mathbf{D}_k \mathbf{U}_k^{\mathrm{t}} \tag{3.45}$$

where $\mathbf{U}_k$ is an upper triangular matrix (with unit diagonal elements) and $\mathbf{D}_k$ is a diagonal matrix with real positive elements. Such a decomposition is called a U-D factorization or square-root factorization, although the computation of square roots is not required. The U-D factorization procedure guarantees non-negativity of the $\mathbf{P}_k$ covariance matrix. The time updating is performed on $\mathbf{U}_k$ and $\mathbf{D}_k$. In this thesis we shall employ the basic RLSK algorithm described here but with U-D factorization of $\mathbf{P}_k$, and shall refer to it as the Sqaure-Root-Kalman (SRK) algorithm. It is identical to the algorithm referred to as the Square-Root-Kalman in [16]. Appendix C lists the computational steps needed to realize the algorithm on a computer, the extensive mathematical manipulations needed to prove them being in [16]. At start-up, the matrices $\mathbf{D}_{-1}$ and $\mathbf{U}_{-1}$ are each equal to the unit identity matrix, and $\underline{\mathbf{c}}_{-1}$ is an all-zero vector.

From Appendix C we see that the number of computations per iteration is proportional to M^2. As observed in the next section, the input vector $\underline{\mathbf{x}}_{k+1}$ contains a number of elements of the previous vector $\underline{\mathbf{x}}_k$ "shifted" along. Algorithms that take advantage of this shifting property are the Fast Kalman algorithm [3], [65] and the Fast Transversal Filter (FTF) algorithms [93], these schemes having the number of computations per iteration now only proportional to M. They are found, however, to be more numerically unstable than the RLSK algorithm of (3.42) [16], [24]. Another algorithm based on minimization of the least square error, and which also takes advantage of the shifting property, is the RLS adaptive lattice algorithm [3], [24], [62], [64], [71], which uses lattice filters to implement the equalizer instead of digital transversal ones. It has the advantage of being order recursive, i.e. an additional new stage to the lattice filter does not alter the state of the other stages, unlike digital transversal filters whose coefficients all change with a different number of taps. It is also more numerically stable than the RLSK algorithm of (3.42). Like the Fast Kalman and FTF algorithms, its complexity is proportional to M, which in this case is the number of lattice stages, but with a significantly larger proportionality factor. For this reason, as pointed out in [16] for the DFE, the SRK algorithm may be more efficient in terms of computations than the RLS adaptive lattice algorithm for typical HF channels. "Normalized" versions [24], [93] of FTF and RLS adaptive lattice algorithms have improved numerical stability, but require the computation of square roots, the RLS adaptive lattice needing more. Where digital precision is limited, special "restart" or "rescue" schemes are required to mitigate the effect of numerical errors, these being most needed in the Fast Kalman and FTF algorithms [93], [95], [96]. It is stated in [95], [96] that the SRK algorithm of [16] produces the most numerically stable version of the RLS adaptive transversal filter. In terms of performance, neglecting numerical errors, the RLSK, Fast Kalman and RLS adaptive lattice algorithms are all similar and equally fast converging [62], as one would expect since they minimize the same cost function.

3.3 Application to Equalizer Adaptation

The vectors $\underline{x}_k$, $\underline{c}_k$ and scalar z_k take the following forms for the DFE and MLSE equalizer structures.

3.3.1 The DFE(N,g)

The conventional approach is to have

$$\underline{x}_k^{\mathrm{t}} = [r_k \; r_{k+1} \cdots r_{k+N-1} \; \hat{s}_{k-1} \; \hat{s}_{k-2} \cdots \hat{s}_{k-g}] \tag{3.46}$$

$$z_k = \hat{s}_k \tag{3.47}$$

where $M = N + g$. It can be verified by taking expectations in (3.5) that

$$\underline{c}_{k,\mathrm{opt}}^{\mathrm{t}} = [w_{k0} \; w_{k1} \cdots w_{k,N-1} \; -b_{k1} \; -b_{k2} \cdots -b_{kg}] \tag{3.48}$$

where $\{b_{ki}\}$ and $\{w_{ki}\}$ are as given in (2.133) and (2.137) of chapter 2. The vector $\underline{c}_k$ is a better estimate of $\underline{c}_{k,\mathrm{opt}}$ than $\underline{c}_{k-1}$ is, but it is $\underline{c}_{k-1}$ that has to be used as the estimate of $\underline{c}_{k,\mathrm{opt}}$ in detecting $\hat{s}_k$.

3.3.2 Channel Estimation

For the MLSE (using the original VA) we need to know the sampled impulse response of the resultant channel as seen by the decision device in fig. 2.5; i.e. the vector

$$\underline{\mathbf{y}}_k^{\mathrm{t}} = [y_{k0}\ y_{k1}\ \cdots\ y_{kg}] \tag{3.49}$$

(Note: do not confuse the $(g+1)$-component vector $\underline{\mathbf{y}}_k$ with the N-component vector $\underline{\mathbf{y}}_{ki}$ in (2.129)). The adaptation problem is therefore one of channel estimation, requiring

$$\underline{\mathbf{x}}_k^{\mathrm{t}} = [\hat{s}_{k-q}\ \hat{s}_{k-q-1}\ \cdots\ \hat{s}_{k-q-g}] \tag{3.50}$$

$$z_k = r_{k-q} \tag{3.51}$$

Vector $\underline{\mathbf{c}}_{k,\mathrm{opt}}=\underline{\mathbf{y}}_{k-q}$, and $\underline{\mathbf{c}}_{k-1}$ is the best available estimate of $\underline{\mathbf{y}}_k$ for use by the MLSE when processing the received sample r_k. Obviously the longer the delay q is, the more inaccurate is the estimate of $\underline{\mathbf{y}}_k$ for a time-varying channel. Predictive methods have been investigated in [66] as a means for improving the channel estimate over significant delays. Another strategy is to use "soft" decisions for s_{k-i}, $0\leq i\leq g$, enabling us to set $q=0$ in (3.50) and (3.51). This can take the form of obtaining a "soft" $\hat{s}_k$ according to some cost function criterion in the MLSE, or, at the expense of greater complexity, having a DFE produce it. For a near-MLSE we effectively already have a DFE anyway, because the digital pre-filter $\mathcal{W}(f)$ required by it is approximated (at low noise and large enough N) by the feedforward section, the feedback section plus $b_{k0}=1$ representing the $(g+1)$-length sampled impulse response seen by the near-MLSE.

3.3.3 Training/Decision-Directed Mode

It will be noticed that $\underline{\mathbf{x}}_k$ depends on the data symbols, in which case the adaptation algorithm can be called *data-aided*. In the start-up of the adaptation process the receiver knows the transmitted data symbols (receiver is in *training* mode), normal transmission only beginning when convergence has been achieved to the extent that receiver decisions are reliable enough to be used (receiver now in *decision-directed* mode). Notice that in decision-directed mode for a rectangular QAM constellation there are four possible solutions to $\underline{\mathbf{c}}_{k,\mathrm{opt}}$, these corresponding to our detecting $\hat{s}_k$ as

$$\hat{s}_k = s_k\, e^{j\pi i/2} \qquad i=0,\ 1,\ 2 \text{ or } 3 \tag{3.52}$$

For i=0 the training mode solution is obtained. If it ever arises that $i \neq 0$ in (3.52), there is no problem as long as the information on which quadrant (in the complex plane) the k^{th} data symbol is in is encoded as the quadrant change between the s_{k-1} and s_k transmitted symbols, and not the actual quadrant in which s_k is in (e.g. see [42]).

Chapter 4
THE HF CHANNEL MODEL

In this thesis we shall adopt the guassian-scatter model, described in [4], to represent a voiceband HF ionospheric channel. As stated in [4], the model can accurately represent a major portion of typical HF ionospheric links, exceptions being those channels that have a specular (non-fading) path present. Experimental verification of the model is given in [67]. For a detailed description of the characteristics of the HF medium, see [2].

In this chapter we shall describe the gaussian-scatter model referred to above, and explain how it has been applied and implemented in this thesis. The discussion will deal with the basic assumptions regarding the models of the two-path and three-path channels tested, and the length of time over which readings are taken. We shall also discuss the problem of carrier phase recovery.

4.1 The Gaussian-Scatter Model

Suppose there are P paths present, and that a path i has associated with it a fixed delay τ_i relative to the first or earliest path. Fig. 4.1 depicts the model of the i^{th} path. For a signal $\mathrm{Re}\{x(t)e^{j2\pi f_c t}\}$ at the input (as in fig. 2.1), the signal at the output is given by $\mathrm{Re}\{G_i(t)x(t-\tau_i)e^{j2\pi f_c(t-\tau_i)}\}$. The complex function $G_i(t)$ is given by

$$G_i(t) = \tilde{G}_{ia}(t)e^{j2\pi\nu_{ia}t} + \tilde{G}_{ib}(t)e^{j2\pi\nu_{ib}t} \tag{4.1}$$

where the "a" and "b" subscripts identify the two magnetoionic components (whose difference in delay is negligible) that are generally present in each path. The quantities $\tilde{G}_{ia}(t)$ and $\tilde{G}_{ib}(t)$ are independent complex gaussian ergodic random processes, each with zero mean and independent real and imaginary components with equal variances. The statistical variation of $G_i(t)$ is independent with respect to each path. The amplitudes of $\tilde{G}_{ia}(t)$ and $\tilde{G}_{ib}(t)$ have Rayleigh probability distribution functions, to give rise to what is commonly called *Rayleigh fading*. The quantities ν_{ia} and ν_{ib} represent the Doppler frequency shifts for the two

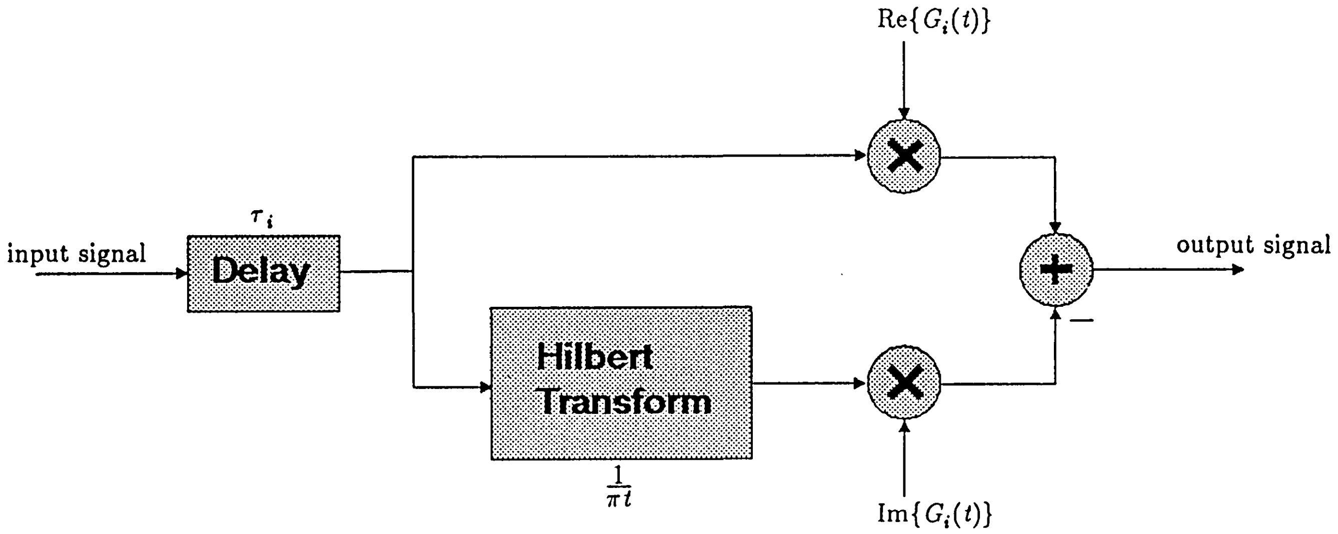


Fig. 4.1 Model of i^{th} path of HF channel.

magnetoionic components. The power spectrums of random processes $\tilde{G}_{ia}(t)$ and $\tilde{G}_{ib}(t)$ are gaussian functions centred on zero. It will be assumed here that $\nu_{ia}=\nu_{ib}=\nu_i$, and that the variances of the gaussian power spectrums of the two magnetoionic components are equal. Then we can write for $G_i(t)$

$$G_i(t) = \tilde{G}_i(t)e^{j2\pi\nu_i t} \tag{4.2}$$

$$F_G(f) = \frac{1}{A_i\sqrt{2\pi}\ f_r}e^{-(f-\nu_i)^2/(2f_r^2)} = \text{FT of E}[G_i^*(x)G_i(x+t)] \tag{4.3}$$

where $F_G(f)$ is the power spectrum of $G_i(t)$ and A_i the attenuation. The quantity $2f_r$ is the Doppler (or frequency) spread of $\tilde{G}_i(t)$, which we have assumed to be approximately the same for each path; it represents the rate with which $\tilde{G}_i(t)$ changes with time, and will often be referred to as the *fade rate*. From [4], it is quite valid to describe $G_i(t)$ by (4.2) and (4.3) for some low path rays, and often also for high rays where the two magnetoionic components have such large differences in delay that they can effectively be treated as distinct paths.

A method for estimating HF channel parameters like Doppler spread, multipath spread, Doppler shift and signal-to-noise ratio, in real-time, is given in [68]. The estimated parameters can then be used in numerous ways in adaptive HF communication setups, including real-time channel quality evaluation for link or equipment adaptation [68].

In chapter 2, fig. 2.1, the transmission medium's causal impulse response was denoted as $h_o(\tau)$, which does not imply any time variation in the transmission characteristic. Let us now denote it as $h_o(\tau;t)$, the t variable expressing its time variation property. From fig. 4.1 we can write

$$h_o(\tau;t) = \sum_{i=0}^{P-1}\text{Re}\{G_i(t)\ [\delta(\tau-\tau_i) - \frac{1}{j\pi(\tau-\tau_i)}]\ \} \tag{4.4}$$

where it is assumed $\tau_0=0$. From (2.8) and (2.13),

$$y(\tau;t) = [a(\tau)\bullet(h_o(\tau;t)e^{-j2\pi f_c(\tau+t_1)})]e^{-j\phi(t)} \tag{4.5}$$

Notice that we have modified ϕ to be a possibly time-varying phase difference $\phi(t)$. The FT of $[\delta(\tau)-1/(j\pi\tau)]$ is a step function of height 2 starting at the origin. Then since the FT of real waveform $a(t)$ is zero beyond the carrier frequency f_c,

$$y(\tau;t) = e^{-j\phi(t)} \sum_{i=0}^{P-1} a(\tau-\tau_i)G_i(t)e^{-j2\pi f_c(\tau_i+t_1)} \tag{4.6}$$

For simplicity we assume $\nu_i=\nu$ for all paths, to give

$$y(\tau;t) = e^{-j\phi'(t)} \sum_{i=0}^{P-1} a(\tau-\tau_i)\tilde{G}_i(t) \tag{4.7}$$

where

$$\phi'(t) = \phi(t)-2\pi\nu t \tag{4.8}$$

the complex constant $e^{-j2\pi f_c(\tau_i+t_1)}$ being incorporated into $\tilde{G}_i(t)$ without any loss of generality (time delays τ_i and t_1 are taken to be fixed). The z-transform of the sampled waveform of impulse response $\sqrt{T}\ y(\tau;t)$, which we defined in (2.113) and (2.117), but without a subscript k to denote time dependence, is

$$\Omega_k Y(z) = \sum_{i=0}^{g} y_{ki} z^{-i} = e^{-j\phi'_k} \sum_{i=0}^{g} \sum_{m=0}^{P-1} \sqrt{T}\ a(iT+\Delta-\tau_m)\tilde{G}_{km} z^{-i} \tag{4.9}$$

where ϕ'_k and $\tilde{G}_{km}$ are the values of $\phi'(t)$ and $\tilde{G}_m(t)$ respectively at the k^{th} sampling instant, and $0 \leq \Delta < T$. We will now digress a little to talk about the phase term $\phi'(t)$.

4.2 Carrier Phase Recovery

In adaptive equalization we need to keep up with the time variations of $\phi'(t)$ and $\tilde{G}_i(t)$. The phase difference $\phi(t)$ between the receiver and transmitter carriers can vary with time because of instabilities in the local oscillators. The frequency shift (or offset) ν can sometimes, under disturbed ionospheric conditions, be as high as 10 Hz, while on "good" or "quiet" days it is typically 0.01–1 Hz [2]. Apart from $\phi'(t)$, each individual path i is subject to an additional phase change given by the time-varying argument of $\tilde{G}_i(t)$. The fade rate of $\tilde{G}_i(t)$ is typically ~1 Hz or less [2], [64], but can be as high as 10 Hz on some transauroral and transequatorial paths [64]. If the term $e^{-j\phi'(t)}$ varies much faster than $\tilde{G}_i(t)$, it is better to have a separate adaptive process for the purpose of tracking $\phi'(t)$. This is because an adaptive digital filter (this includes a channel estimator as well) is mainly meant to compensate for, and therefore track, the distortion introduced by the channel's frequency spectrum characteristic (which leads to ISI), as governed by the $\{\tilde{G}_i(t)\}$ terms in (4.7). The extra burden of tracking $e^{-j\phi'(t)}$, particularly if varying at a much faster rate, can seriously degrade performance [69]. The quantity $\phi'(t)$ is a phase change that is common to all paths, and can therefore be thought of as an "effective" phase difference between the transmitter and receiver carriers. Thus the task of following $\phi'(t)$ is commonly referred to as *carrier phase recovery* or *tracking.* The linear nature of QAM permits the tracking of $\phi'(t)$ using quite accurate data-aided techniques, without needing to send an auxillary transmitted pilot tone.

Let our estimate of ϕ'_k in (4.9) be denoted as $\tilde{\phi}'_k$. Then in the case of the DFE(N,g) [46] the output of the feedforward section is multiplied by $e^{j\tilde{\phi}'_{k-1}}$, which is equivalent to multiplying the samples r_{k+i}, $0 \leq i \leq N-1$, in the feedforward section by $e^{j\tilde{\phi}'_{k-1}}$, so that $\underline{\mathbf{x}}_k$ in (3.46) is modified as

$$\underline{\mathbf{x}}_k^t = [r_k e^{j\tilde{\phi}'_{k-1}} \dots r_{k+N-1} e^{j\tilde{\phi}'_{k-1}} \ \hat{s}_{k-1} \dots \hat{s}_{k-g}] \tag{4.10}$$

The update $\tilde{\phi}'_k$ is given by an SD algorithm that seeks to adjust $\tilde{\phi}'_k$ so that the $\mathrm{E}[|\epsilon_k|^2]$ is minimized, i.e.

$$\tilde{\phi}'_k = \tilde{\phi}'_{k-1} + \mu \mathrm{Im}\{\epsilon_k [e^{j\tilde{\phi}'_{k-1}} \sum_{i=1}^{N} r_{k+i-1} c_{k-1,i}]^*\} \tag{4.11}$$

$\tilde{\phi}'_k$ is a better estimate of ϕ'_k than $\tilde{\phi}'_{k-1}$ is, but we have to use $\tilde{\phi}'_{k-1}$ as the best available estimate of ϕ'_{k+i}, $0 \leq i \leq N-1$, in the detection of $\hat{s}_k$.

It can be seen that the update $\underline{c}_k$ involves $\tilde{\phi}'_{k-1}$ whilst $\tilde{\phi}'_k$ involves $\underline{c}_{k-1}$, so that we have a kind of joint adaptation of $\underline{c}_k$ and $\tilde{\phi}'_k$ going on. Note that for minimization of $\mathrm{E}[|\epsilon_k|^2]$ the joint optimum choice of the feedforward tap coefficients and phase $\tilde{\phi}'_k$ at the k^{th} instant is not unique, and can theoretically take on an infinite number of combinations. For example, if $\tilde{\phi}'_{k,\mathrm{opt}} = \phi'_k + \psi$, where ψ is any real value, then the optimum feedforward tap weights are given by those optimum tap weights for $\tilde{\phi}'_{k,\mathrm{opt}} = \phi'_k$ multiplied by $e^{-j\psi}$. Therefore, this means that $\tilde{\phi}'_k$ is not necessarily a literal estimate of ϕ'_k, but should be looked on rather as a quantity that follows (with μ in (4.11) sufficiently high) the same rapid changes in ϕ'_k that the slower tracking digital filters cannot accommodate. Note that in [46] carrier demodulation is also carried out along with the carrier phase compensation. This is merely for convenience, and should make no difference. Hence in [46] one has what is termed a "passband" DFE, as opposed to the baseband one considered here, because demodulation of the carrier frequency is performed after the feedforward section.

In channel estimation for the MLSE we multiply each received sample r_{k-q} by $e^{j\tilde{\phi}'_{k-1}}$, so that (3.51) becomes

$$z_k = r_{k-q} e^{j\tilde{\phi}'_{k-1}} \tag{4.12}$$

and the update $\tilde{\phi}'_k$ is given by

$$\tilde{\phi}'_k = \tilde{\phi}'_{k-1} - \mu \mathrm{Im}\{\epsilon_k [e^{j\tilde{\phi}'_{k-1}} r_{k-q}]^*\} \tag{4.13}$$

The MLSE at the k^{th} instant operates directly on r_k using $e^{-j\tilde{\phi}'_{k-1}}\underline{c}_{k-1}$ as an estimate of $\underline{y}_k$.

For further details and analysis of carrier phase recovery, see [46], [52], [53], [69], [70].

4.3 Channel Configurations Tested

This thesis is concerned with the adaptation of the digital receiver filter, the desired function of which is to mitigate the distortion caused by the $\{\tilde{G}_i(t)\}$ terms in $y(\tau;t)$ (4.7). Therefore, as most other contributions on the subject also do (e.g. [11], [14]–[18], [64], [66], [71]), we shall exclude the variation of $e^{-j\phi'(t)}$ from our investigations; this is equivalent to assuming $e^{-j\phi'(t)}$ has a negligible time variation, so that it can effectively be incorporated as a constant in $\tilde{G}_i(t)$, or that it is perfectly compensated for by a carrier phase recovery scheme. The CCIR [72] also recommend that for general qualitative testing of a system of data transmission on simulated HF circuits, under good/moderate/poor conditions, a two-path channel model be used with equal mean attenuation, equal fade rates, and no Doppler shifts for each path; assuming stable carrier oscillators, this then means that $e^{-j\phi'(t)}$ is time-invariant.

With $\phi'(t)=0$ the resultant channel impulse response seen by the receiver is

$$y(\tau;t) = \sum_{i=0}^{P-1} a(\tau-\tau_i)\tilde{G}_i(t) \tag{4.14}$$

and the sampled impulse response (4.9) is

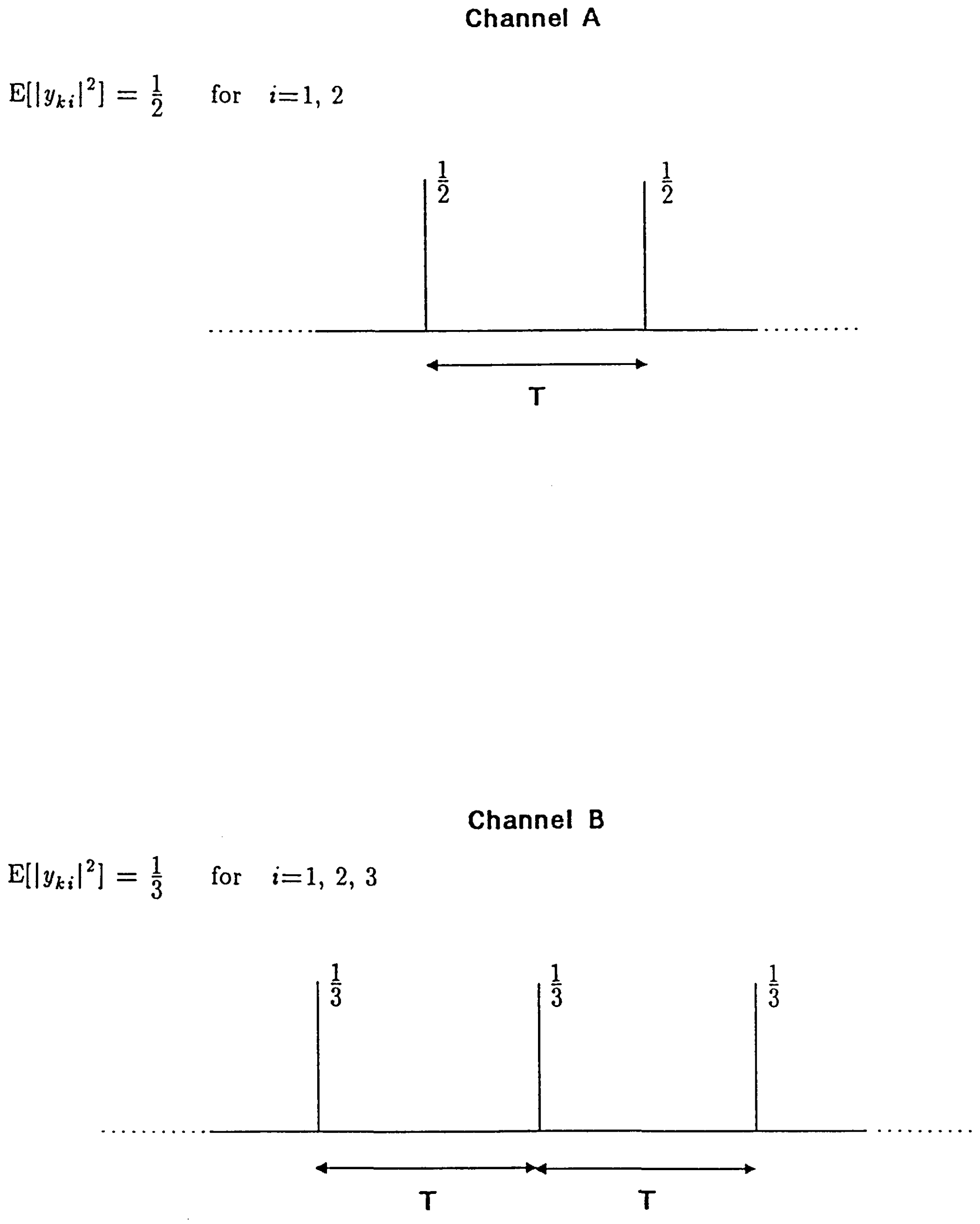
$$\Omega_{kY}(z) = \sum_{i=0}^{g} y_{ki} z^{-i} = \sum_{i=0}^{g}\sum_{m=0}^{P-1} \sqrt{T}\; a(iT+\Delta-\tau_m)\tilde{G}_{km} z^{-i} \tag{4.15}$$

Each component y_{ki} is a complex gaussian random process with fade rate $2f_r$, the real and imaginary parts being statistically independent of each other with zero means and equal variances. The correlation between components is

$$\begin{aligned} \mathrm{E}[y_{ki} y_{kh}^*] &= \sum_{m=0}^{P-1}\sum_{n=0}^{P-1} T\,\mathrm{E}[\tilde{G}_{km}\tilde{G}_{kn}^*] a(iT+\Delta-\tau_m) a^*(hT+\Delta-\tau_n) \\ &= \sum_{m=0}^{P-1} \frac{T}{A_m} a(iT+\Delta-\tau_m) a^*(hT+\Delta-\tau_m) \end{aligned} \tag{4.16}$$

It can be seen that the correlation is dependent on the shape of the data pulse $a(t)$ and time delays $\{\tau_i\}$. A simplification frequently made is to assume that y_{ki}, if of non-zero variance, is statistically independent of y_{kh}, $h\neq i$, (true if $a(t)\approx 0$ for $t<0$, $t\geq T$, i.e. no interference between adjacent pulses) and of equal variance; see, for example, [14]–[16], [64], [71]. This simplification will also be made here. Each non-zero y_{ki} then represents the contribution from a distinct transmission path, all the paths contributing equally in terms of average attenuation and fade rate.

It was mentioned in chapter 1 that the number of "effective" paths present in an HF channel is generally small. In [12], for example, a 1000 km HF link is shown to have 2–4 paths present in general; the channel can effectively be modelled as a three-path channel [55]. The investigations to be reported in this thesis are based on two configurations for $\Omega_{kY}(z)$, denoted as Channel A and Channel B, and depicted in fig. 4.2. Channel A is representative of a two-path channel with a time delay between paths of about T secs., and is similar to the two-path channels used in [14]–[16], [64]. Channel B is representative of a three-path channel with a multipath spread of about $2T$ secs., and is similar to the three-path channels used in [14], [64], [71]. In terms of multipath spread, the channels can be classed as being "good", in the sense



Channel A

$E[|y_{ki}|^2] = \frac{1}{2}$ for $i=1, 2$

Channel B

$E[|y_{ki}|^2] = \frac{1}{3}$ for $i=1, 2, 3$

Fig. 4.2 Channel configurations.

that the spread is kept to a minimum for the given number of resolvable paths.

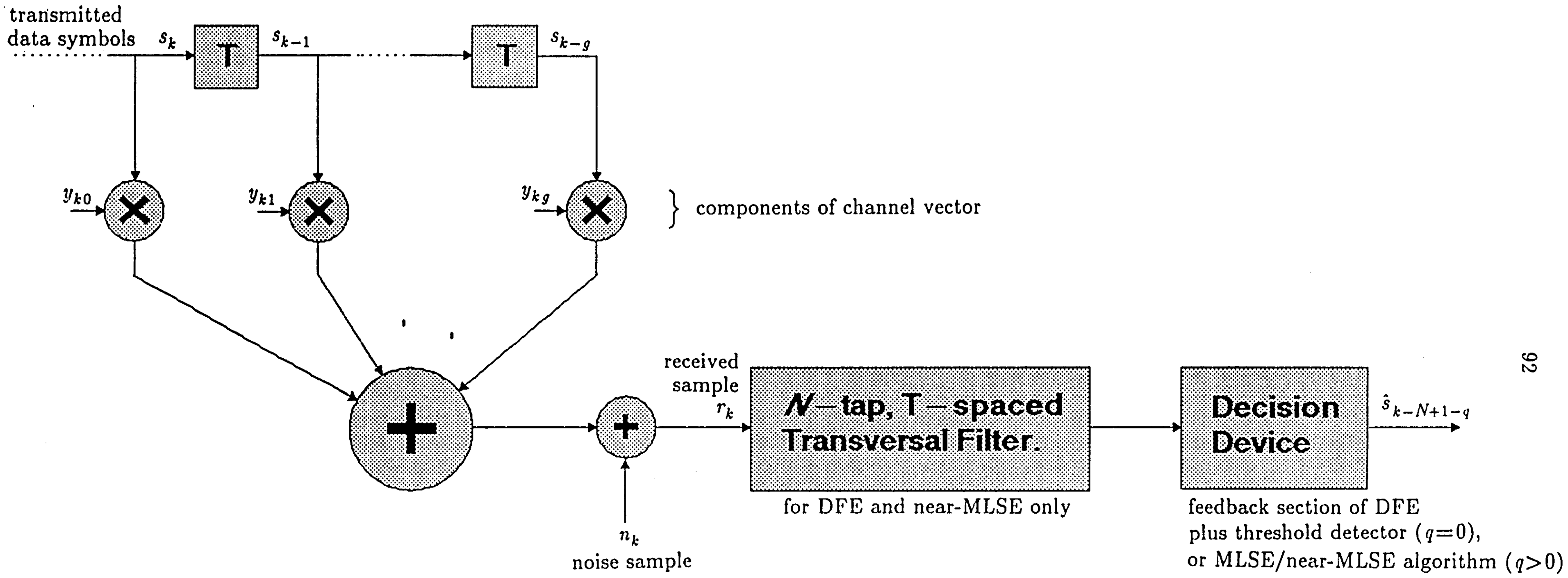
For midlatitude paths the fade rate is typically between 0.1 and 1.0 Hz [64], with more extreme fading in the range 1.0 to 10.0 Hz being encountered on certain transauroral and transequatorial paths. Most simulation studies concerning transmission over HF channels have assumed fade rates of 2.0 Hz or less, the most commonly used being 1.0 Hz. In this thesis we shall examine performance primarily at the highest fade rate for midlatitude paths, namely 1.0 Hz, and to a lesser extent we will also look at performance at a fade rate of 2.0 Hz. Furthermore, the performance in mode I, where adaptation is assumed perfect, can equivalently be viewed as performance at a very low fade rate, since the tracking algorithm is then able to follow the channel's variations with negligible error. Thus, altogether, three different fade rates will be examined. In future the abbreviation f.r. will sometimes be used for "fade rate", and should not be confused with f_r; the relationship is f.r.$=2f_r$.

Note that although it is assumed T^{-1}=2400, the results of our investigations could equally well be interpreted as being for T^{-1}=1200, with the fade rates of 1.0 Hz and 2.0 Hz then becoming 0.5 Hz and 1.0 Hz respectively. The multipath spreads would also be twice as large.

4.4 Notes on Implementation

4.4.1 Model of the Transmission System

The basic model of the transmission system, as implemented on the computer, is shown in fig. 4.3. An outline of the simulation process is as follows. At each new time instant kT the data symbols are shifted along in the shift register shown in fig. 4.3, and a new symbol s_k generated at the input. The channel vector $\underline{y}_k$ and noise sample n_k are generated, in a manner to be described shortly, and the multiplication and addition process carried out to produce the received sample r_k. The contents of the transversal receiver filter are shifted along, and the new sample r_k accepted at the input. The output of the transversal filter (which is simply r_k if the MLSE is being used) is fed to the decision device, whereupon an estimate $\hat{s}_{k-N+1-q}$ of the



N.B. all quantities are complex

Fig. 4.3 Model of the Transmission System.

symbol $s_{k-N+1-g}$ is produced. The tracking algorithm (not indicated in fig. 4.3) then adjusts the appropriate parameters in the receiver, ready for the next time instant $(k+1)T$.

The simulation programs, given in Appendix I, are written in Fortran-77 and were run on a Cyber 170-720 mainframe computer, which uses 60 bit words with 48 bits (equivalent to approximately 14 decimal digits) for the mantissa.

4.4.2 Signal-to-Noise Ratio

The k^{th} sample at the input to the transversal receiver filter is

$$r_k = \sum_{i=0}^{g} s_{k-i} y_{ki} + n_k \tag{4.17}$$

and the signal-to-noise ratio is therefore

$$SNR_k = \frac{\sigma_s^2}{\sigma_n^2} \sum_{i=0}^{g} |y_{ki}|^2 \tag{4.18}$$

The average signal-to-noise ratio is

$$SNR = \mathrm{E}[SNR_k] = \frac{\sigma_s^2}{\sigma_n^2} = \frac{1}{\rho} \tag{4.19}$$

where, as shown in fig. 4.2, the variance of the $\{y_{ki}\}$, $0 \leq i \leq g$, are normalized so that their sum is unity, i.e

$$\mathrm{E}[|y_{ki}|^2] = \frac{1}{(g+1)} \qquad \text{for} \quad 0 \leq i \leq g \tag{4.20}$$

In practice the receiver employs an automatic-gain-control (AGC) amplifier to bring the level of the received signal into its dynamic range, e.g. see [16].

For convenience of layout of the graphs in this thesis, the definition of signal-to-noise ratio as given in (4.19) is adopted. For comparing different data constellation sizes it is more fair, however, to divide *SNR* by the number of bits per symbol. Hence it should be remembered that for obtaining the signal-to-noise ratio per bit from *SNR* (dB) the values should be adjusted as follows:

(*i*) 64 QAM ≡ 6 bits per symbol; therefore subtract approximately 8 dB.

(*ii*) 16 QAM ≡ 4 bits per symbol; therefore subtract approximately 6 dB.

(*iii*) 4 QAM ≡ 2 bits per symbol; therefore subtract approximately 3 dB.

4.4.3 Generation of $\{y_{ki}\}$, $\{n_k\}$ and $\{s_k\}$

The series of values $\ldots\, y_{k-1,i}\,,\ y_{ki}\,,\ y_{k+1,i} \ldots$ are samples of a complex gaussian ergodic random process with zero mean and independent real and imaginary components with equal variances. The power spectrum of the random process is gaussian with variance f_r^2. The inverse FT of $(\sqrt{2\pi}\, f_r)^{-1} e^{-f^2/(2f_r^2)}$ is $e^{-2(f_r\pi t)^2}$; hence the correlation between samples is

$$\mathrm{E}[y_{ki}y^*_{hi}] = \frac{1}{(g+1)} e^{-2(f_r\pi(k-h)T)^2} \tag{4.21}$$

Consider the T-spaced samples given by

$$y_{ki} = \sum_m n_{mi} f(kT-mT_1) \tag{4.22}$$

where $\{n_{mi}\}$ are rate T_1^{-1} samples of a complex gaussian ergodic random process with zero mean and independent real and imaginary components with equal variances. The power spectrum of the random process is constant (process is therefore white) with value σ_1^2, so that

$$\mathrm{E}[n_{mi}n^*_{hi}] = \sigma_1^2\delta_{mh} \tag{4.23}$$

The quantity $f(t)$ is a real function. Since the $\{y_{ki}\}$ as defined by (4.22) are linear functions of $\{n_{mi}\}$, it follows that they too are samples of a complex gaussian ergodic random process with zero mean and independent real and imaginary components with equal variances. Denoting the FT of $f(t)$ as $F(f)$, the correlation between samples is

$$\begin{aligned}\mathrm{E}[y_{ki}y^*_{hi}] &= \sum_m \sigma_1^2\, f(kT-mT_1)\, f^*(hT-mT_1) \\ &= \sigma_1^2 \sum_m \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u) e^{j2\pi u(kT-mT_1)} F^*(w) e^{-j2\pi w(hT-mT_1)}\, du\, dw\end{aligned}$$

$$= \sigma_1^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u)F^*(w)e^{j2\pi(uk-wh)T}\sum_m e^{j2\pi(w-u)mT_1}\, du\, dw \tag{4.24}$$

Using the relation

$$\sum_m e^{j2\pi(w-u)mT_1} = \frac{1}{T_1}\sum_m \delta(w-u-\frac{m}{T_1}) \tag{4.25}$$

we obtain

$$\mathrm{E}[y_{ki}y^*_{hi}] = \frac{\sigma_1^2}{T_1}\int_{-\infty}^{\infty} F(u)\sum_m F^*(u+\frac{m}{T_1})\; e^{-j2\pi hmT/T_1}\; e^{j2\pi u(k-h)T}du \tag{4.26}$$

Assume T_1 is such that

$$F(u)F^*(u+\frac{m}{T_1}) \approx 0 \quad \text{for} \quad m\neq 0 \tag{4.27}$$

Then

$$\mathrm{E}[y_{ki}y^*_{hi}] \approx \frac{\sigma_1^2}{T_1}\int_{-\infty}^{\infty} |F(u)|^2\; e^{j2\pi u(k-h)T}du \tag{4.28}$$

Clearly $(\sigma_1^2/T_1)|F(f)|^2$ is the power spectrum of the random process of which $\{y_{ki}\}$ are samples. In order to satisfy (4.21) we then require that

$$\frac{\sigma_1^2}{T_1}|F(f)|^2 = \frac{1}{(g+1)\sqrt{2\pi}\, f_r}e^{-f^2/(2f_r^2)} \tag{4.29}$$

Choosing $\sigma_1^2 = \frac{1}{(g+1)}$ means we can have

$$F(f) = \sqrt{\frac{T_1}{\sqrt{2\pi}\, f_r}}\; e^{-f^2/(4f_r^2)} \tag{4.30}$$

Therefore $$f(t) = \sqrt{2\sqrt{2\pi}\; T_1 f_r}\; e^{-(2f_r\pi t)^2} = \sqrt{\frac{T_1}{\sqrt{\pi}\; T_2}}\; e^{-t^2/(2T_2^2)} \tag{4.31}$$

where $$T_2 = \frac{1}{2\sqrt{2}\; f_r\pi} \tag{4.32}$$

In order to satisfy (4.27) we need

$$e^{-(2f_rT_1)^{-2}} = e^{-2\pi^2(T_2/T_1)^2} \ll 1 \tag{4.33}$$

If we select $$T_1 = 0.4T_2 \tag{4.34}$$

then (4.33) becomes

$$e^{-12.5\pi^2} \approx 2.64 \times 10^{-54} \ll 1 \tag{4.35}$$

and $f(t)$ is

$$f(t) = \frac{1}{\sqrt{2.5\sqrt{\pi}}} e^{-t^2/(2T_2^2)} \tag{4.36}$$

To sum up, the rate T^{-1} samples $\{y_{ki}\}$ representing the time variation of the i^{th} component of the channel sampled impulse response can be generated by passing white gaussian samples $\{n_{ki}\}$, at rate T_1^{-1}, through a filter $f(t)$ of impulse response given by (4.36) (with $T_2 = 2.5T_1$).

In practice we shall approximate $f(t)$ by the waveform depicted in fig. 4.4, this waveform agreeing with (4.36) at the 13 sample points shown. Linear interpolation is used between the sample points. The relationship between T_1/T and f_r is (from (4.32) and (4.34))

$$\frac{T_1}{T} = \frac{1}{5\sqrt{2}\,\pi f_r T} \tag{4.37}$$

If we select T_1 to be a multiple of T, then the clocking period of the $\{n_{ki}\}$ samples is a multiple of the baud period, which provides for easier program implementation. With $T^{-1} = 2400$, $T_1/T = 216 \Rightarrow f_r \approx 0.5001757$ Hz and $T_1/T = 108 \Rightarrow f_r \approx 1.000351$ Hz, which clearly shows that accurate fade rates of 1.0 Hz and 2.0 Hz can be achieved.

The method of generating the $\{y_{ki}\}$ that has been described here is a more refined version of the one described in [73], where the accuracy of the method is demonstrated using $\{n_{ki}\}$ that are simple binary sequences and approximations to $f(t)$ that are much cruder than that in fig. 4.4.

Each of the elements in the channel impulse response vector $\underline{y}_k$ has its own independent generating sequence $\{n_{ki}\}$, $0 \le i \le g$. It will be noticed that, apart from a multiplying constant, the random sequence $\{n_{ki}\}$ is statistically equivalent to the noise sample sequence $\{n_k\}$ in (4.17). Thus the method of producing the noise samples $\{n_k\}$ and the $\{n_{ki}\}$ samples for the $\{y_{ki}\}$, $0 \le i \le g$, is the same, and is described in Appendix D. It employs the computer's own random generating function which, for any initial starting "seed", produces a sequence of real

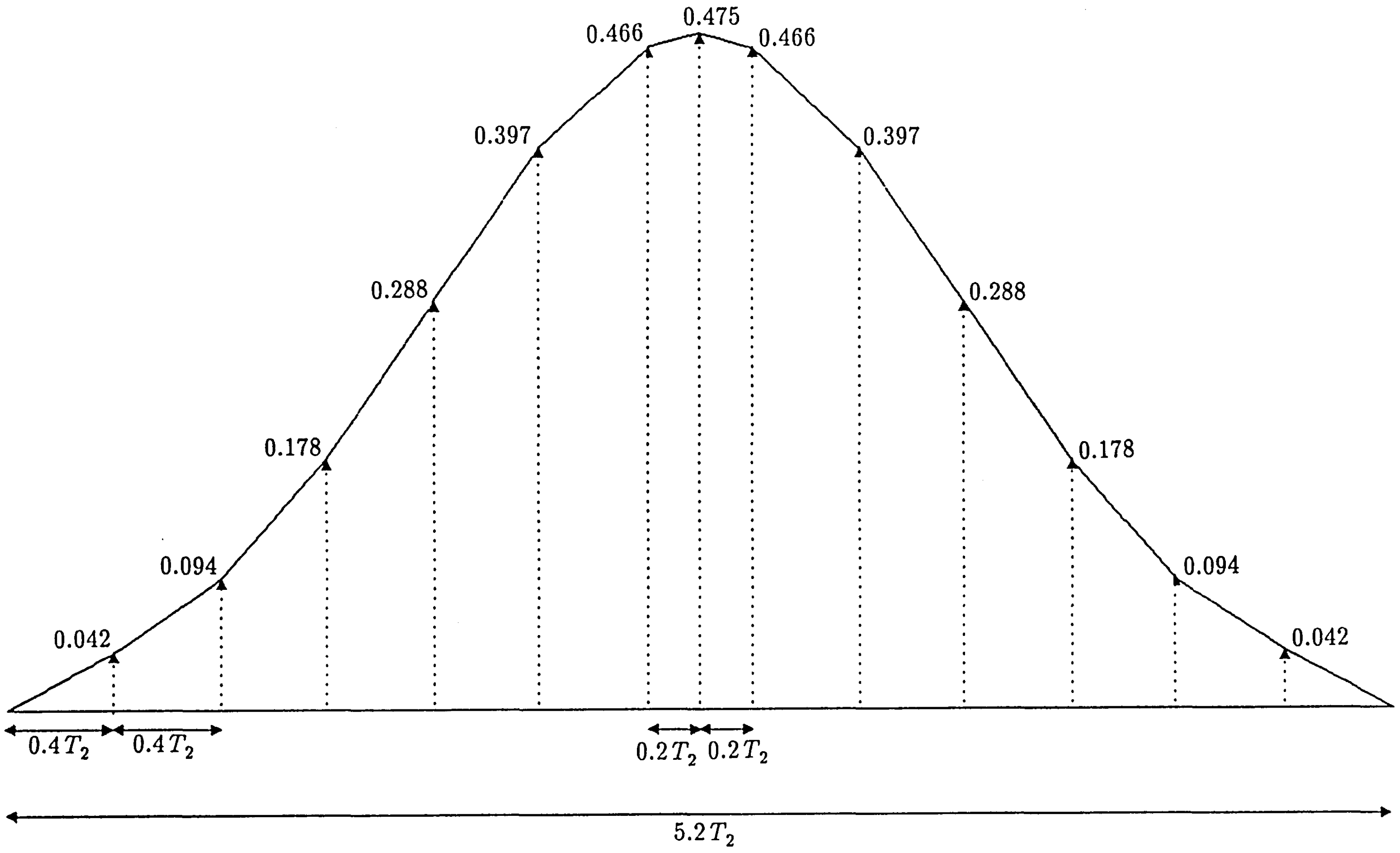


Fig. 4.4 Approximation to Gaussian Function $\left(\sqrt{2.5\sqrt{\pi}}\right)^{-1}.e^{-t^2/(2T_2^2)}$

numbers varying between 0 and 1 with a uniform probability distribution. Each number in a sequence is statistically independent from any other, and different starting seeds produce independent sequences.

The method of generating a sequence of statistically independent data symbols $\{s_k\}$ is also based on the computer's random generating function. The range 0 to 1 is divided into L^2 equal sub-intervals, so that a random real number falling into any particular sub-interval designates a particular data symbol to be transmitted.

4.5 Time Span Observed

To gauge performance of the different detection techniques using tracking algorithms, it is necessary, for practical reasons, to make measurements from some limited observation time of the received signal. Figs. 4.5 and 4.6 depict a typical variation with time of the norm $\|\underline{\mathbf{y}}_k\|^2$ of the channel impulse response vector $\underline{\mathbf{y}}_k$, for Channel A and Channel B respectively. The time span is 25 fade periods. The graphs are indicative of the fading pattern of the instantaneous signal-to-noise ratio SNR_k (4.18). The norm $\|\underline{\mathbf{y}}_k\|^2$ for Channel B is $\frac{2}{3}$ times that for Channel A, plus the contribution from an extra component, $|y_{k2}|^2$.

In this thesis all measurements conducted over Channel A and Channel B are based on the respective fading patterns in figs. 4.5 and 4.6. For a fade rate of 1.0 Hz and 2.0 Hz, the 25 fade period time span corresponds to the transmission of 60,000 and 30,000 data symbols respectively. In the next chapter the effects on performance measure of using a limited observation time will be discussed briefly. The sequences of 60,000 and 30,000 data symbols and noise samples are the same for all measurements, unless specifically stated otherwise, with the 30,000-sequences being the first 30,000 of the 60,000-sequences.

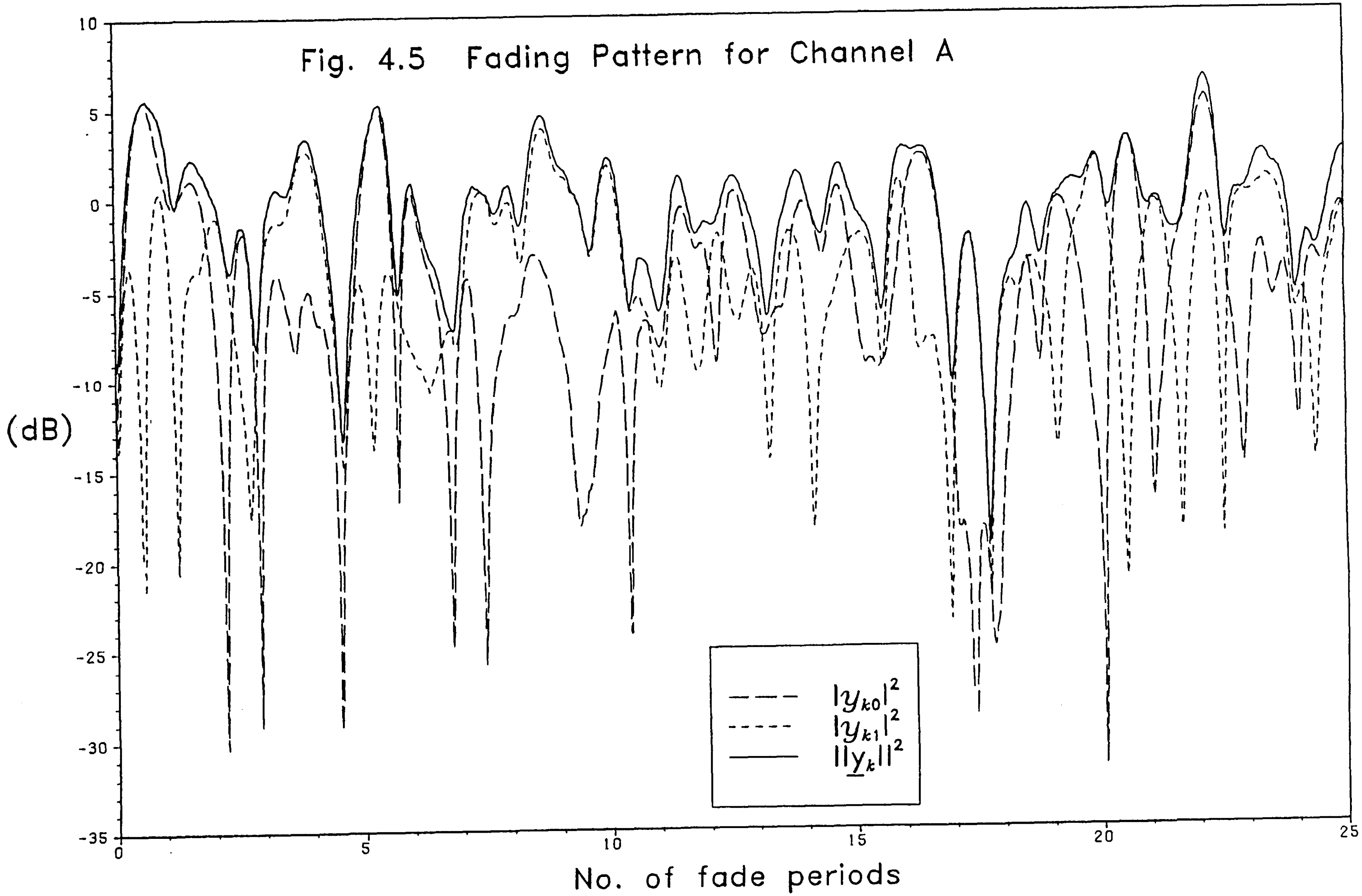


Fig. 4.5 Fading Pattern for Channel A

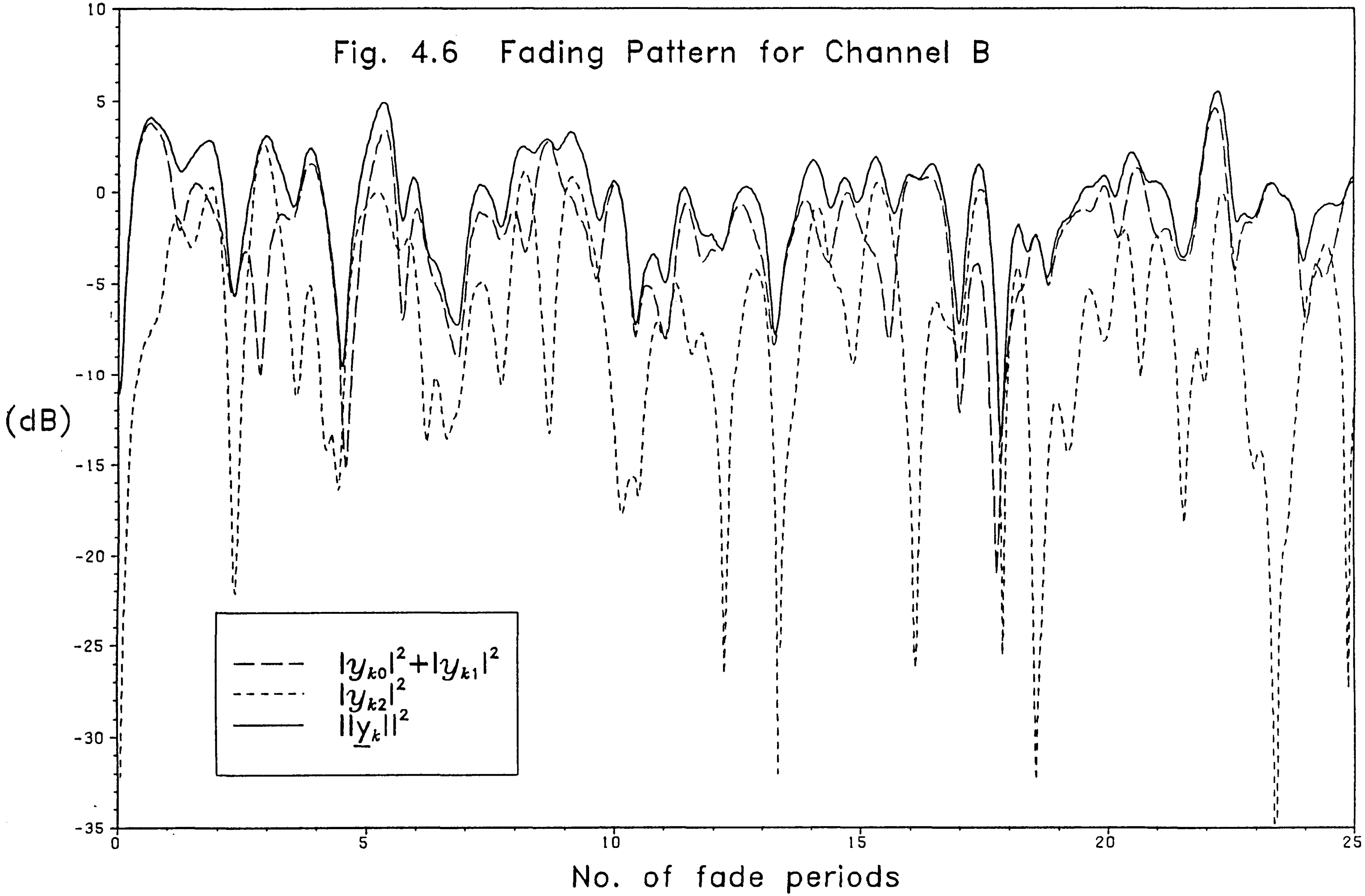


Fig. 4.6 Fading Pattern for Channel B

Chapter 5
MODE I PERFORMANCE

In this thesis we will examine three modes of detection at the receiver. These are:

mode I: receiver always knows channel state perfectly, i.e. perfect tracking assumed. This is equivalent to having a very slow fading channel, such that the error in adaptation is negligible.

mode II: tracking algorithm uses correct data symbols at all times, i.e. receiver decisions not allowed to affect adaptation.

mode III: tracking algorithm uses receiver decisions on data symbols. This, of course, represents the situation to be encountered in practice.

In addition, mention of modes I and II may sometimes bear a subscript of "1" or "2" that has relevance to the DFE only. Subscript "1" refers to the situation where the feedback process uses correct data symbols, thus eliminating any propagative error effect in receiver decisions. Subscript "2" refers to the situation where the feedback process uses previous receiver decisions. Comparison of performance in situation "1" with that in situation "2" will reveal the effect of error propagation in the DFE. As the number of decision errors in situation "1" gets smaller, the performance in situation "2" should tend to that in situation "1". In mode III the feedback process of the DFE uses receiver decisions always. Frequently we will use an abbreviation like say DFE-I_1, which denotes operation of the DFE in mode I_1.

Mode I tells us the "ideal" performance one can expect from a particular decision device, and this chapter is going to look at this aspect. We shall look at the performance of the ZF MFE, and compare this with the DFE(N,g), with N taking on various values. This will reveal how far the finite-tap DFE is from the very best that can be expected. The DFE is also compared with the MLSE/near-MLSE and merits discussed. To begin with, however, we will talk a little about the performance criteria that will be used in this and subsequent chapters.

5.1 Performance Criteria

5.1.1 Probability of Error

5.1.1.1 Threshold Detection, as used in the DFE and MFE

The two most popular criteria for performance evaluation in the detection of data are the mean square error and the probability of error, the latter by far being the more informative. In the presence of gaussian noise and no ISI the probability of error in the detection of a data symbol can be calculated quite straightforwardly.

A difficulty arises when ISI is present due to the fact that a closed form expression for the probability distribution function (p.d.f.) of the ISI is almost impossible to obtain. This is further compounded in a time-varying environment because the p.d.f. of the ISI is then constantly changing. More problems arise in the feedback process of a DFE because it can lead to propagative effects should a decision error occur. Consequently virtually all error rate performance evaluations over HF links (and some time-invariant channels too) in the literature use the brute force approach of actually counting the number of errors that have occurred and dividing by the total number of symbols transmitted. A drawback with this method is that to obtain error estimates below ~1 in 10^4 usually entails the transmission of quite long sequences of symbols, and so rarely is the observed error probability below 10^{-4}.

The probability of error in modes I_2, II_2 and III, because of the difficulty in analysing the effect of propagation, will be examined using the error-count approach. Modes I_1 and II_1, however, can be handled using an analytical method that estimates the error probability using the statistics of the noise and ISI. Such a procedure enables us in theory to measure the error rate up to 1 in 10^4 and beyond. We will now mention a few of the methods available for measuring the probability of error in modes I_1 and II_1.

It is possible to get an exact expression for the probability of error in the presence of ISI using a series expansion technique [74], but the method can be rather involved. The hard effort in obtaining an exact answer has naturally led to the development of more easily calculable bounds on the error rate. Perhaps the simplest of these is the upper bound formulated by Saltzberg [75], which is based on the Chernoff bound. The Saltzberg bound suffers from being too loose in the region of most practical interest [78], [79], a fact confirmed in our own tests (the Saltzberg bound is used in [80] to measure the probability of error on a two-path microwave channel that assumes perfect knowledge of the channel and no propagation in the DFE). Other bounds by Milewski [76] and McLane [77] are tighter but require that the peak distortion, or eye-opening, (defined in Appendix E) be less than 1. This is not always the case in our situation, especially during a deep fade (when tracking will be suffering) and with 64 QAM. The "sharp" upper and lower Chebyshev bounds of [78] were found by us to be very tight when using 4 QAM but extremely loose with 64 QAM. This was most likely due to the increased peak distortion that was present at the higher constellation size.

An attractive set of bounds has been formulated by Jenq, Liu and Thomas [79]. Here a simple lower bound, an upper bound, and a simple approximation to the upper bound which is twice the lower bound, are derived. Thus either the lower bound or the approximation to the upper bound can be taken as a good approximation to the actual probability of error. We have derived the bounds in Appendix E, slightly diffrently from [79], and show that one can obtain a tighter lower bound than that given in [79]. More precisely, the approximation to the upper bound is now only at most twice the lower bound, and we show in Appendix E that the two bounds tend to coincide as the peak distortion decreases, whereas in [79] they always remain 3 dB apart. In this thesis we adopt the upper bound, as given in Appendix E, as our measure for the probability of error. It is evaluated at each time instant *kT* to take into account the time variation of the transmission medium, and then averaged with previous values.

In future we shall denote the probability of error at any instant *kT*, relevant only when we employ the analytical method, by P_{e_k}. The average (over time) probability of error will be denoted as P_e, even when we use the error-count approach, although we shall identify that situation with the abbreviation "E.C.". Besides modes I_1 and II_1, the analytical P_e estimate will prove useful in mode III under certain conditions, as will be seen later in chapter 8.

It is important to mention the distinction between probability of symbol error, as we use in this thesis, and the probability of a bit error, or bit error rate as it is usually called. It is common practice, e.g. [42], to have the original data in the form of binary, and then to code sets of $2\log_2 L$ bits into the L^2-level data symbols that are to be transmitted. Thus, although we can say that the bit error rate will be lower than the symbol error rate, its precise value is dependent on the coding scheme used.

5.1.1.2 MLSE/near-MLSE

The probability of error for an MLSE, as implemented by the VA, is also very difficult to evaluate exactly, perhaps more so than the DFE. Bounds on the error rate, which can be fairly tight, have been developed [3], [23], [30], [33] assuming perfect knowledge of the channel impulse response, which implies the only error in the cost evaluation process of the true survivor comes from gaussian noise. The evaluation of these bounds can still be quite involved, relying on the determination of a "minimum distance" quantity that also changes as the channel medium does. When there is imperfect knowledge of the channel impulse response, the error in the cost evaluation process of the true survivor now also involves contributions from data symbols. In [34] an attempt to upper bound performance with this added degradation is made by application of the Chernoff bound.

The storage and computational requirements of implementing an MLSE using 16 and 64 QAM force us to adopt a sub-optimum near-MLSE scheme, which uses an adaptive pre-filter. The prospect of estimating the probability of error by analysis now becomes even more daunting. In view of all the difficulties mentioned, we shall therefore measure error rate performance by the error-count approach for the MLSE and near-MLSE.

5.1.2 Mean Square Error

In modes I_1 and II_1 there are two ways of calculating the average (over time) mean square error in detection in the DFE. The first, which is more costly, is that knowing the true channel state and the DFE taps we can calculate what the coefficients of the interference terms from data symbols are. Adding together the squared magnitude of each coefficient and multiplying by σ_s^2 gives us the mean square interference from data symbols, and then adding this to the variance of the noise (equal to the sum of the squared magnitudes of each feedforward tap multiplied by σ_n^2) gives us the mean square error at any instant, ξ_k. Averaging ξ_k over time gives us the average mean square error. The second way is to simply subtract the true symbol at the k^{th} instant, s_k, from the pre-threshold estimate $\tilde{s}_k$, and then average the squared magnitude of this error over time. Over a long enough time interval the two methods should produce the same result, since the data symbols, noise and channel state all vary independently, and over the 25 fade period time spans used here this was found to be virtually so, the difference being negligible. In this thesis we have used the first method, at no great cost, since the calculation of the interference terms is already required in the evaluation of the analytical estimate P_{e_k}.

In future we shall denote the average mean square error as ξ. However, we will not use this criterion as much as the more interesting probability of error, restricting its use to the DFE in modes I_1 and II_1. In the MLSE there is no directly comparable performance measure to ξ, since there is no analagous quantitiy to a pre-threshold estimate of a data symbol.

5.2 ZF MFE(g+1)

In chapter 2 it was shown that the best performance, in terms of probability of error, is achieved with a ZF MFE, in which the interference from data symbols is zero. From (2.150) and Appendix E we can write the probability of error at instant kT as

$$P_{e_k} \approx 2(1-\tfrac{1}{L})\text{erfc}\Big(\frac{\|\underline{y}_{k0}\|}{\sigma_n}\Big) \tag{5.1}$$

where $\underline{y}_{k0}$ is given by

$$\underline{y}_{k0}^{t} = [y_{k0}\ y_{k+1,1} \cdots y_{k+g,g}] \tag{5.2}$$

Vector $\underline{\mathbf{y}}_{k0}$ should not be confused with vector $\underline{\mathbf{y}}_k$, which is given by

$$\underline{\mathbf{y}}_k^{\mathrm{t}} = [y_{k0}\ y_{k1} \cdots y_{kg}] \tag{5.3}$$

although for a slowly fading channel c.f. baud rate it would not be unsafe to assume $\underline{\mathbf{y}}_{k0} \approx \underline{\mathbf{y}}_k$. Notice that the argument of the erfc(.) term in (5.1) is the reciprocal of the square root of ξ_k, the mean square error in detection at time kT.

The average probability of error, P_e, and the average mean square error, ξ, are formed by averaging samples of P_{e_k} and ξ_k respectively over the fading patterns. Figs. 5.1–5.4 show P_e and ξ for a ZF MFE averaged over 30,000 uniformly spaced samples on Channel A and Channel B. P_e is the lowest obtainable error rate, since it is for a ZF MFE with perfect knowledge of the channel state and with correct decisions on past and future data symbols. ξ for a ZF MFE is not the minimum achievable error, but as the noise power decreases it does tend toward the average MSE obtained from an MMSE MFE (2.149) as shown in the figures.

It is possible to derive P_e and ξ for a ZF MFE from the assumed statistics of the fading channel. This is done in Appendix F, from which we get (using (F.14) and (F.15))

$$P_e = 2\left(1-\frac{1}{L}\right)\left\{1-\sqrt{\frac{\lambda}{1+\lambda}}\left(1+\frac{1}{2(1+\lambda)}\right)\right\} \tag{5.4}$$

$$\xi = 2\sigma_s^2\rho \tag{5.5}$$

for Channel A, and

$$P_e = 2\left(1-\frac{1}{L}\right)\left\{1-\sqrt{\frac{\lambda}{1+\lambda}}\left(1+\frac{1}{2(1+\lambda)}+\frac{3}{8(1+\lambda)^2}\right)\right\} \tag{5.6}$$

$$\xi = 1.5\sigma_s^2\rho \tag{5.7}$$

for Channel B, where

$$\lambda = (\rho(g+1)\sigma_s^2)^{-1} \tag{5.8}$$

The theoretical expressions of (5.4)–(5.7) are also plotted in figs. 5.1–5.4, from which we can see the effect of using 25 fade periods as opposed to an infinite number. The difference is negligible in the ξ's, but quite substantial in the P_e's, particularly at the lower error rates. To confirm that the difference is due to the low number of fade periods covered, and not a statistically inaccurate channel model, we also made measurements (shown in figs. 5.1–5.4) using 60,000 uniformly spaced samples over a time span of 5400 fade periods (which is equivalent to a fade rate of 216 Hz with T^{-1}=2400). It can be seen that the agreement with the theoretical P_e curves is much better, although the lower error rate values are still sensitive to the time span used. This was confirmed by observing that measurements using only the first 30,000 samples, which are over a time span of 2700 fade periods, tended to fall below the theoretical P_e curves at the lower error rates rather than above them as in the 5400 fade period case.

The reason why the lower error rates are increasingly sensitive to the number of fade periods covered is that as the level of noise drops, P_e is governed by the more deeper fades. Since the deeper the fade the less frequent its occurrance, we need to have a larger time span containing a sufficient number of them so as to form a good average estimate. The fact that ξ exhibits a much closer fit to theory than P_e for the same time span is due to the performance measure itself; the P_e measure places a much greater weight to the deep fade occurrences.

A perhaps more economical technique to obtaining good long-term estimates of the error rate, rather than observing long continuous cycles of fading, is to measure the error rate at various "snapshot" instants of the random variation of the channel [64]. Each instant should be chosen independently of any other, the average of all of them providing the long-term estimate, and the larger the number of instants the more statistically accurate the estimate. This technique is fine as long as we are assuming perfect knowledge of the channel at each instant. In this thesis we want to observe the performance when decision-directed adaptation is employed, which makes it necessary for us to use a long un-interrupted sequence of received samples. Therefore the "snapshot" technique is unsuitable, and to enable comparison of "ideal" performance with "adaptation" performance is why we are thus making observations over a continuous fading pattern for the former situation.

It will be noticed that Channel B yields a superior performance to Channel A. This is due to the extra path component in Channel B, which makes a deep fade less likely to occur (as can be seen in the fading patterns in figs. 4.5 and 4.6) and consequently a high P_{e_k} and ξ_k less probable. This may also explain why Channel B exhibits a better fit to the theoretical P_e curves than Channel A, because the fade pattern for B is less deviant from the mean. An interesting point from eqs. (5.4)–(5.7) is that for a given ρ, the value of ξ for Channel B is always approx. 1.25 dB lower than the value for Channel A, whereas the difference between the P_e's varies and can be as high as 10 dB or more, as can be seen from figs. 5.1 and 5.3. This suggests that the diversity advantage of Channel B is not expressed in the first-order statistic of ξ_k.

5.3 MMSE DFE(N,g)-I$_1$

Figs. 5.5–5.8 show P_e and ξ for an MMSE DFE(N,g) in mode I$_1$ for various numbers of feedforward taps N. The averages again use 30,000 uniformly spaced samples. The change in P_e and ξ as N varies is more marked for the higher constellation size. This is because the residual mean square ISI in detection is proportional to L^2, and so for a given N and ρ the ISI is more influential the higher the value of L. Increasing N reduces the mean square ISI. The increased coincidence of the curves as N increases suggests that performance close to that with $N=\infty$ can be achieved with a modest number of feedforward taps.

Figs. 5.5 and 5.7 also show the advantage of a ZF MFE over an MMSE DFE. This advantage, in terms of SNR, is given in Table 5.1 for $P_e=10^{-3}$ and 10^{-6} with N at its largest value. Bearing in mind that the P_e curves for the DFE have a lower bound at most 3 dB less, the advantage of the ZF MFE, and therefore also the MLSE, over the MMSE DFE does not appear to be spectacular. A similar observation, using 4 QAM, was made recently in [81] from measurements on real HF channels and using a different performance measure (that of "minimum distance") which also, as here, ignores the effect of propagation in the DFE.

SNR loss (dB) of MMSE DFE c.f. ZF MFE	Channel A		Channel B	
	$P_e=10^{-3}$	$P_e=10^{-6}$	$P_e=10^{-3}$	$P_e=10^{-6}$
4 QAM	1.3	2.0	1.4	1.0
16 QAM	1.9	2.6	1.8	1.1
64 QAM	2.1	3.0	1.8	1.3

Table 5.1 Advantage of ZF MFE over MMSE DFE.

Notice that the advantage of the MFE over the DFE is slightly higher for the larger constellation size, which is due to the increased residual mean square ISI in detection in the DFE. An interesting feature in figs. 5.5–5.8 is that for the same ξ level, at a given N, the corresponding P_e level is higher for Channel A than for Channel B. This suggests that for the same ξ, the time variance of ξ_k is greater for Channel A than Channel B, thus producing a larger error rate. Also, in terms of ξ, the difference between Channel A and Channel B for the DFE(N,g) curves with $N \geq 6$ is quite small, in contrast to the more substantial difference in terms of P_e. This indicates that the ξ criterion does not demonstrate the diversity advantage of Channel B.

Notice from figs. 5.5 and 5.7 the significant advantage in *SNR* an extra path component has on the P_e performance. Roughly speaking, for a P_e of 10^{-3} and 10^{-6} there is a power saving on Channel B of about 4 dB and 6 dB respectively, relative to Channel A.

One might expect that the DFE performance be closer to the MFE performance on Channel A than on Channel B, because of the reduced ISI on average present on A. This is borne out by the ξ curves and the higher error rate part of the P_e curves. The difference (in terms of *SNR*) between the MFE and DFE appears slightly larger for Channel A than Channel B on the low error rate portion of the P_e curves. This is most likely due to the reasons given in the last section, where the lower P_e measurements, unlike the ξ measurements, are more dependent on the actual limited fading pattern covered. This could therefore make them less representative of a truly long-term average value, this likelihood being greater for Channel A.

We shall in future be using a DFE with N=6 on both Channel A and Channel B, this value of N appearing to give a performance that would be reasonably close to that with $N=\infty$ for all constellation sizes without also being unduly large. It should be noted, however, that for a given accuracy we can get away with fewer feedforward taps to approximate to the DFE(∞,g) when using a smaller constellation size.

Fig. 5.9 shows a plot of the ratio of the average mean square noise to the average mean square ISI, these two quantities being the constituents of the average mean square error in detection for the DFE. It is straightforward to show, using eqs. (2.35), (2.55) and (2.59), that for the time-invariant MMSE DFE(∞,g) this ratio is monotonically increasing with decreasing ρ. In the case being studied here we have a finite number of feedforward taps, which means, unlike for the infinite-tap case, that the mean square ISI can never be zero when more than one feedforward tap is non-zero. In this thesis the feedforward section of the DFE(6,g) will be used as an approximation to the feedforward section of the ZF DFE(∞,g) (whose mean square error in detection consists entirely of gaussian noise), as is ideally required by a reduced-state near-MLSE. The graph of fig. 5.9 indicates that for the MMSE DFE(6,g) the gaussian noise dominates the ISI on average; above SNR= 10 dB the noise:ISI ratio is greater than 2.5:1, above 20 dB it is greater than 7:1, and above 27.5 dB it is greater than 10:1.

5.4 MLSE/near-MLSE and MMSE DFE(6,g)

Implementing an MLSE by means of the VA requires that we always hold in store L^{2g} survivor-sequences, and at each new time instant carry out $L^{2(g+1)}$ cost evaluations and comparisons, as detailed in chapter 2. Using 4 QAM, implementation of the VA is not a great problem, but with 16 and 64 QAM it becomes rather expensive. We have therefore decided to implement a near-MLSE when using these larger constellations. The sub-optimum scheme chosen, which is a type of reduced-state VA, is one that has been developed in [42] for high data rate transmission using 64 and 256 QAM over telephone circuits, and is referred to as

"Detector 2" in [42]. If the detector can operate satisfactorily with 64 QAM, its performance should be good with 16 QAM as this is just a subset of the former. The detection process is described in Appendix G. We have constrained it to operate with 16 survivors that are "expanded" into a maximum of 64 sequences during the selection process, these constraints being chosen (from consideration of the results in [42]) to provide a reasonable balance between storage/computational complexity and performance. For details on how performance varies with different constraints over differing quality time-invariant channels, see [42]. Since this detection process is a type of reduced-state VA, we need an adaptive phase filter ahead of it to make the resultant channel response be in a condition of minimum phase. This filter will be approximated by the feedforward section of the MMSE DFE(6,g), with the resultant response at its output assumed (by the detection algorithm) to consist of a unit first component followed by the feedback taps of the DFE.

The delay in detection of the MLSE/near-MLSE is set to 15 symbol intervals for both Channel A and Channel B, this setting being greater than $5g$.

Figs. 5.10 and 5.11 depict the P_e vs. SNR performance of the MLSE/near-MLSE and DFE in mode I. The E.C. curves consist of straight lines between the data points. Tables 5.2 and 5.3 give the number of errors occurring in the E.C. measurement runs, which were done using a 1 Hz fade rate and 60,000 transmitted symbols. As expected the E.C. curves for the DFE-I_1 are quite close to the analytical upper bound estimate, the E.C. values beyond about 1 in 10^4 tending to deviate from the bound. This is because of the inaccuracy of using only a few errors to produce a P_e estimate, a value of 10^{-4} being given by only 6 errors.

The errors occurring in mode I_1 are potential pitfalls for error propagation in the DFE in mode I_2. With 4 QAM the difference between modes I_1 and I_2 is fairly small. For example, from figs. 5.10 and 5.11 at the SNR for $P_e = 10^{-3}$ in mode I_1, the effect of propagation increases the error rate by a factor of approximately 1.3 and 1.8 for Channel A and Channel B respectively. As the number of errors in mode I_1 decreases, the number in mode I_2 will approach it. With

SNR (dB)	No. of errors in 60,000		
	DFE(6,1)		MLSE/ near-MLSE
4 QAM	Mode I_1	Mode I_2	
7.5	4,657	5,730	4,788
10.0	2,139	2,768	2,219
12.5	912	1,186	931
15.0	400	548	416
20.0	92	136	78
24.0	20	21	18
27.5	4	5	6
30.0	1	1	2
16 QAM			
15.0	6,905	10,379	8,998
20.0	1,429	2,649	1,880
25.0	264	458	405
30.0	61	160	89
32.5	22	51	24
35.0	8	16	10
37.5	2	7	7
64 QAM			
25.0	2,711	7,017	5,485
30.0	507	1,245	921
35.0	120	311	249
37.5	54	220	157
40.0	18	77	27
42.5	9	76	17
45.0	1	9	0

Table 5.2 Error counts for mode I on Channel A.

SNR (dB)	No. of errors in 60,000		
	DFE(6,2)		MLSE/
4 QAM	Mode I_1	Mode I_2	near-MLSE
7.5	4,647	6,609	5,154
12.5	800	1,316	748
16.5	109	208	107
20.0	22	34	16
22.5	5	8	3
16 QAM			
17.5	3,558	8,340	6,097
22.5	420	1,486	908
26.5	44	101	59
30.0	6	16	3
31.0	3	3	0
64 QAM			
25.0	3,113	13,872	9,250
30.0	338	3,812	1,222
32.5	76	736	160
35.0	22	77	45
37.5	2	74	0

Table 5.3 Error counts for mode I on Channel B.

the higher constellation sizes the degradation caused by propagation is more severe. For example, from figs. 5.10 and 5.11 with 16 QAM, at the *SNR* for $P_e=10^{-3}$ in mode I_1, the effect of propagation increases the error rate by a factor of approximately 2.7 and 2.4 for Channel A and Channel B respectively, and with 64 QAM the corresponding factors are approximately 3.8 and 7.3. This is due to the fact that there are more levels in the data symbols, for if supposing we were in an error propagation situation then, at worst, if we guessed each symbol at random we would be 25% of the time right for 4 QAM and 6.3% and 1.6% right for 16 and 64 QAM respectively, which suggests it would be less easier to recover out of a propagation run with a higher constellation size. As another example, observe in Table 5.3 that with 4 QAM at *SNR*=22.5 dB, 5 errors in mode I_1 lead to only 8 errors in mode I_2, whereas with 64 QAM at *SNR*=37.5 dB just 2 errors in mode I_1 lead to a staggering 74 errors in mode I_2. Also, note that not all propagation bursts are equally severe, since the lower *SNR* of 35.0 dB with 64 QAM gives 22 errors in mode I_1 (20 more than at 37.5 dB) leading to 77 errors in mode I_2 (just 3 more than at 37.5 dB). This is not surprising since the length of a propagation burst depends on the amount of ISI subtracted out by the feedback taps, and this in turn depends on the random fading of the channel. In general, whenever there are errors in the feedback decisions of the DFE, the noise margin in the detection of the current symbol is significantly reduced. We could therefore expect that the length of a particular propagation burst is also dependent on the noise samples and data symbols that are present at the time. Because propagation bursts have an uncertain duration, when there are a small number of errors for the DFE-I_1 the corresponding errors for the DFE-I_2 may not change in proportion to a change in the number of errors for the DFE-I_1, particularly for a high constellation size, which explains the erratic behaviour around $P_e \lesssim 10^{-3}$ of the DFE-I_2 curves for 64 QAM. Notice that the effect of propagation is generally more severe on Channel B than on Channel A. This is because Channel B is, on average, more dispersive than Channel A, which leads to a larger number of previous symbol decisions being used in the feedback process.

One might expect the MLSE to perform at least as well as, if not better than, the DFE-I_1.

The results for 4 QAM more or less bear this out, with the difference between the two detection devices being very slight. The data given in Tables 5.2 and 5.3 show that for 4 QAM the DFE-I_1 sometimes produces less errors than the MLSE over the fading patterns. The results for 16 QAM, however, show that the near-MLSE is performing overall slightly worse than the DFE-I_1, and for 64 QAM the degradation is more apparent. Thus the MFE advantage detailed in the last section seems to be a poor indicator of the near-MLSE "gain" over the DFE-I_1.

On the other hand, the near-MLSE does perform better than the DFE-I_2. For example, from fig. 5.10 at the *SNR* for $P_e=10^{-3}$ with the MLSE/near-MLSE on Channel A, the relative factors of increase in the error rate given by the DFE-I_2 are approximately 1.6, 1.9 and 2.1 for 4, 16 and 64 QAM respectively, and from fig. 5.11 the corresponding factors for Channel B are approximately 2.0, 1.7 and 2.1. It appears, in fact, that the advantage of the MLSE/near-MLSE over the DFE only arises because of the latter's proneness to error propagation bursts. As pointed out in chapter 2, the MLSE/near-MLSE will also produce errors in bursts, but these are usually much shorter than the propagation bursts of the DFE, as is shown, for example, in [37]. Whether having a near-MLSE, because of its advantage over the DFE-I_2, is worth the considerable amount of extra processing involved is another question.

We shall discuss the MLSE/near-MLSE and DFE devices further in chapter 7, when we look at performance in mode II.

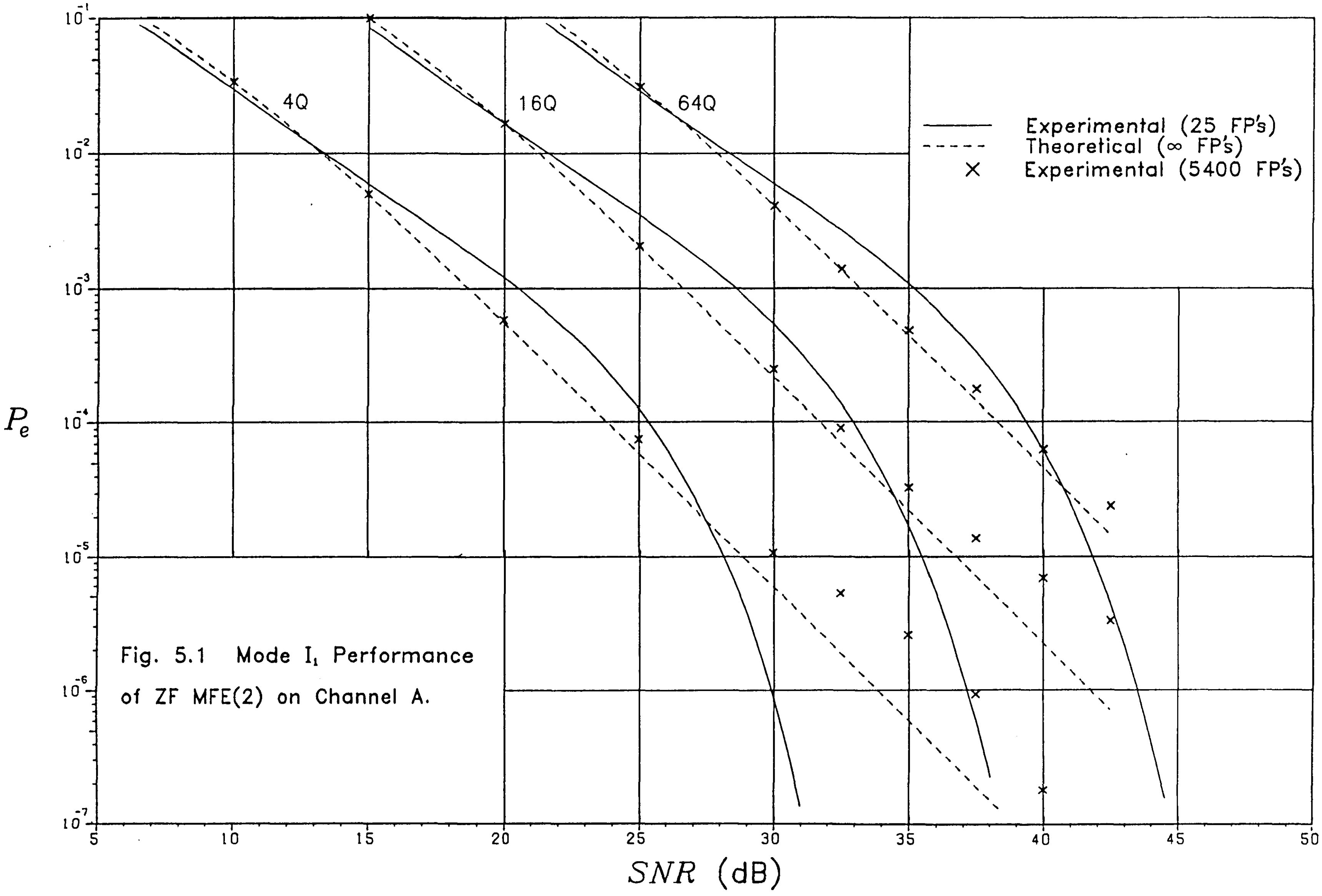
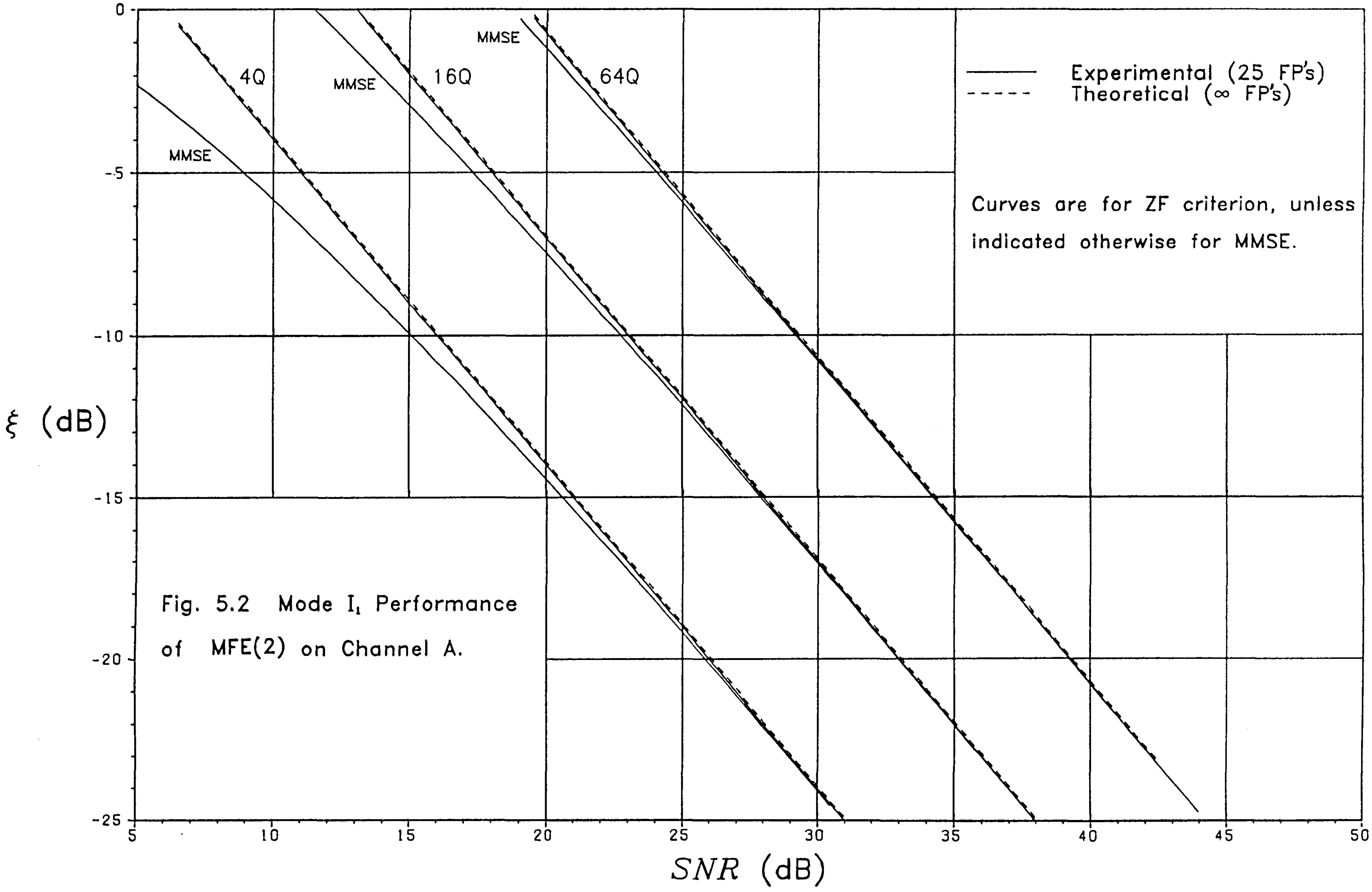


Fig. 5.1 Mode I_1 Performance of ZF MFE(2) on Channel A.



Curves are for ZF criterion, unless indicated otherwise for MMSE.

Fig. 5.2 Mode I_1 Performance of MFE(2) on Channel A.

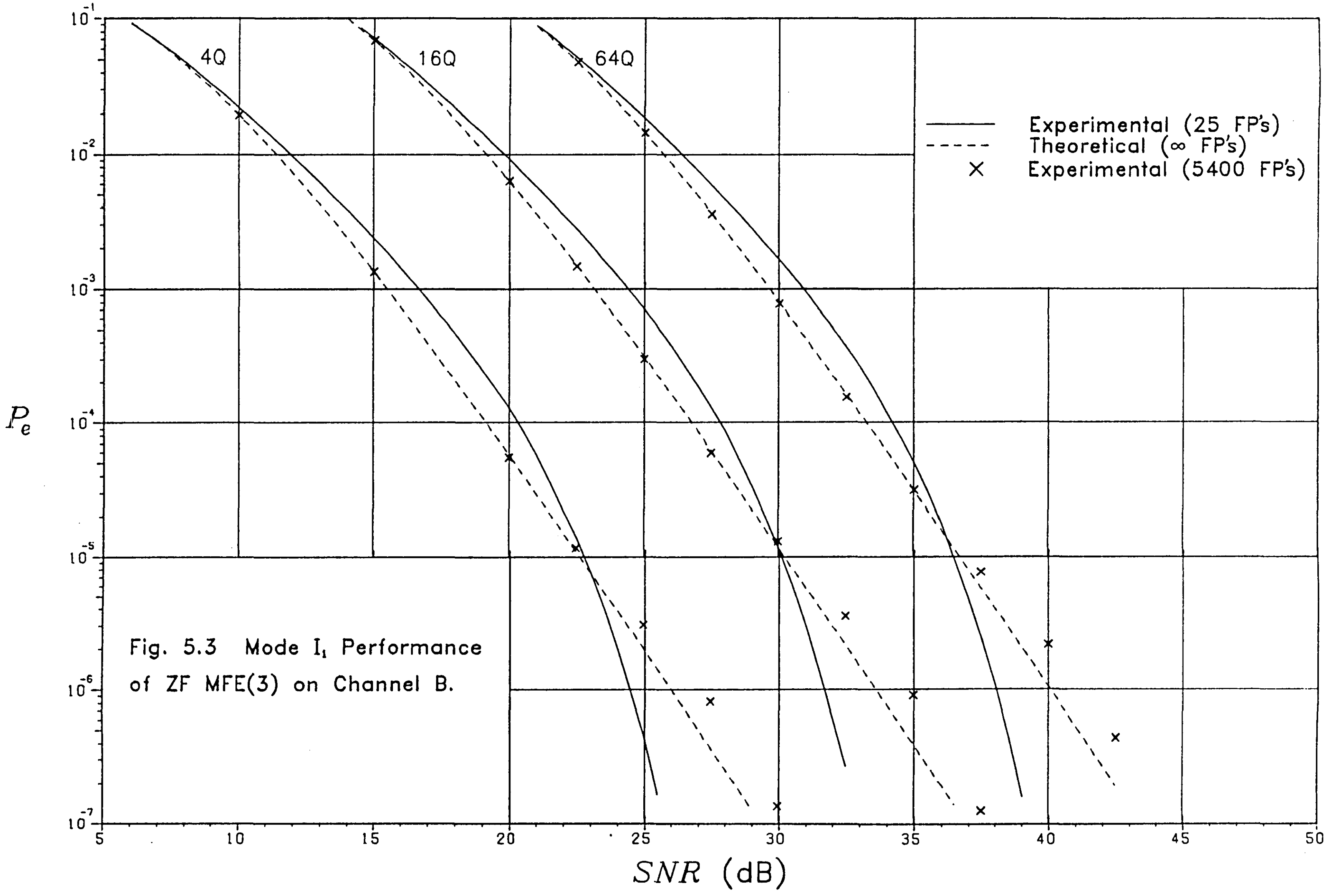
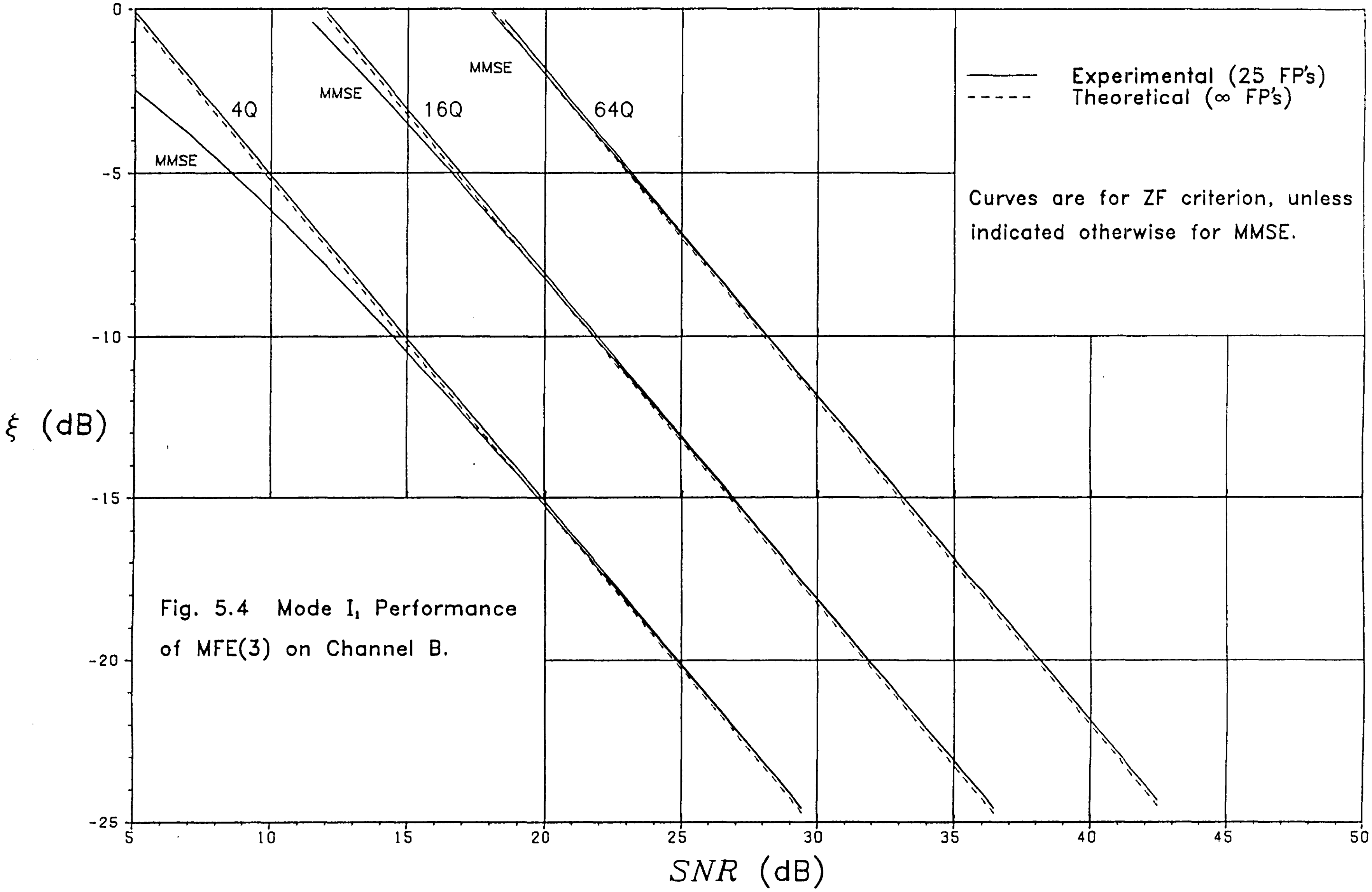
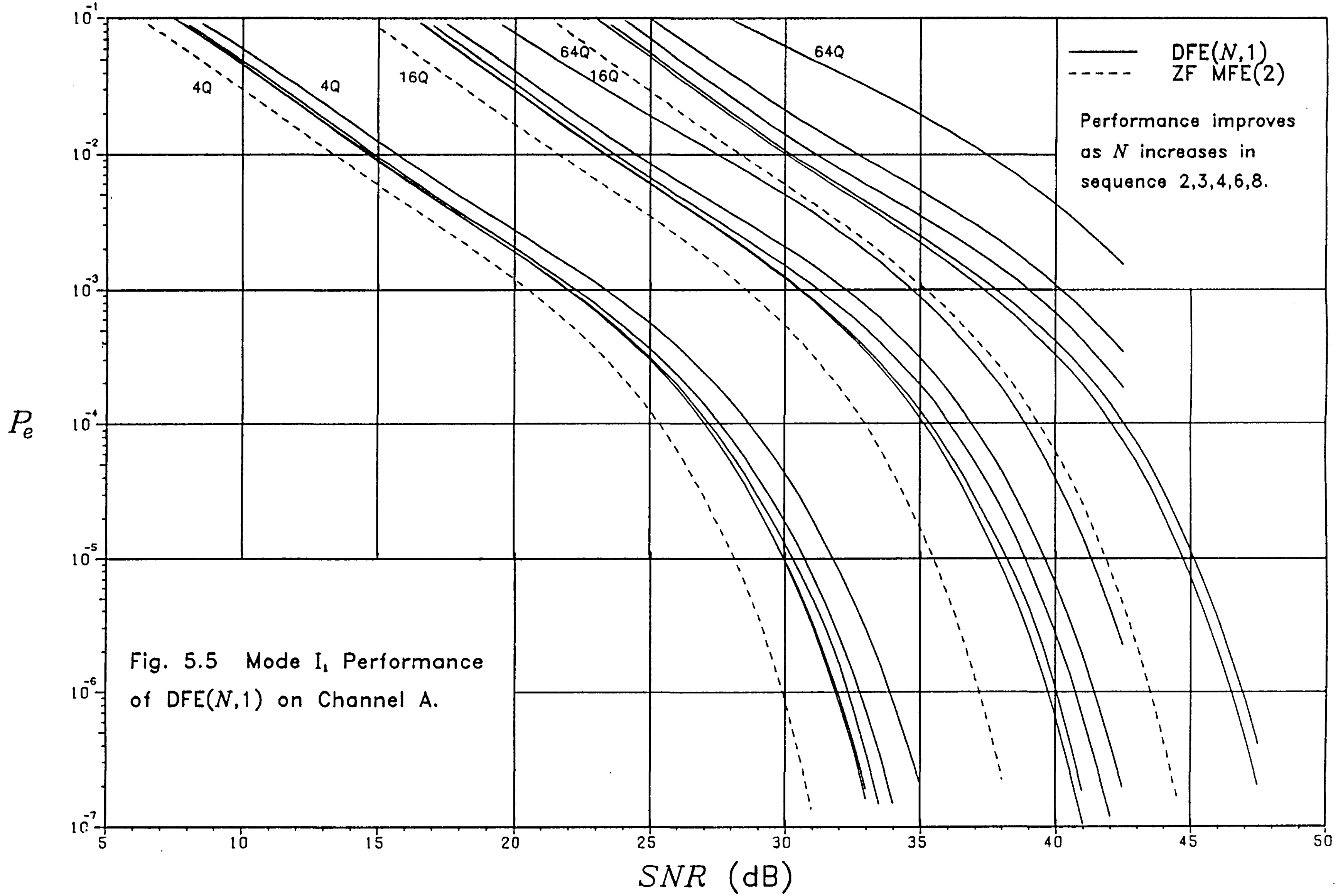


Fig. 5.3 Mode I_1 Performance of ZF MFE(3) on Channel B.



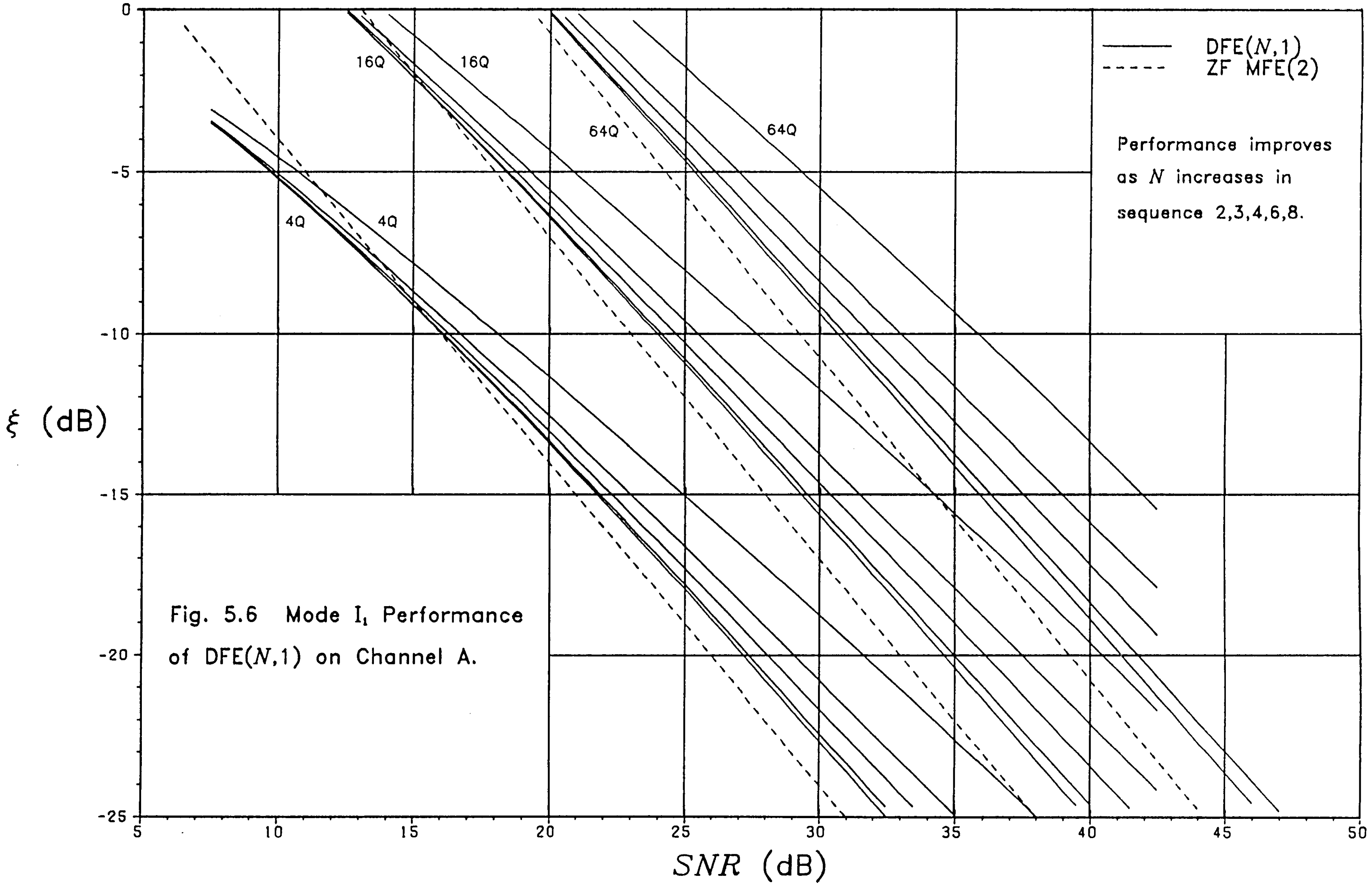
Curves are for ZF criterion, unless indicated otherwise for MMSE.

Fig. 5.4 Mode I_1 Performance of MFE(3) on Channel B.



Performance improves as N increases in sequence 2,3,4,6,8.

Fig. 5.5 Mode I_1 Performance of DFE(N,1) on Channel A.



Performance improves as *N* increases in sequence 2,3,4,6,8.

Fig. 5.6 Mode I_1 Performance of DFE(*N*,1) on Channel A.

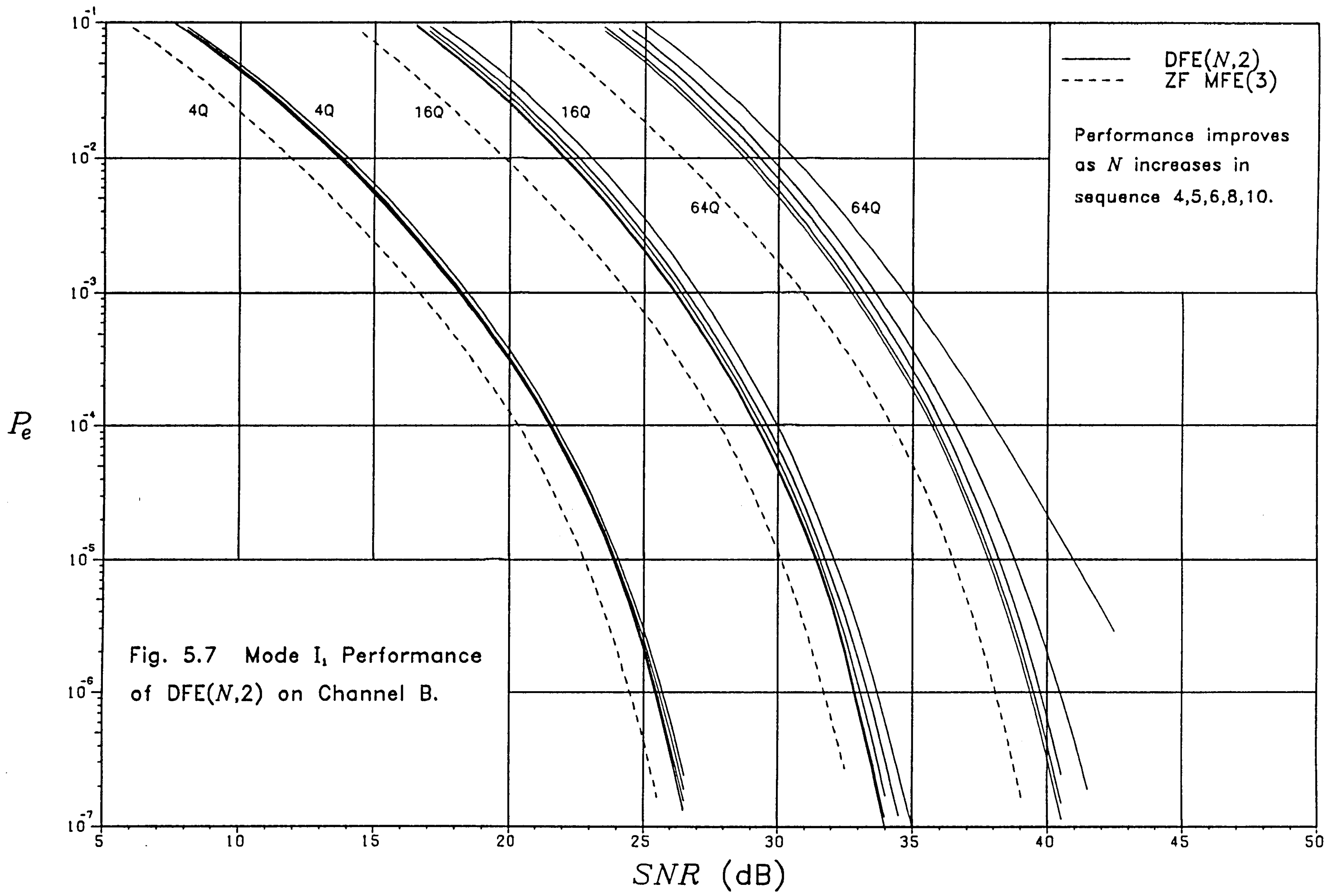
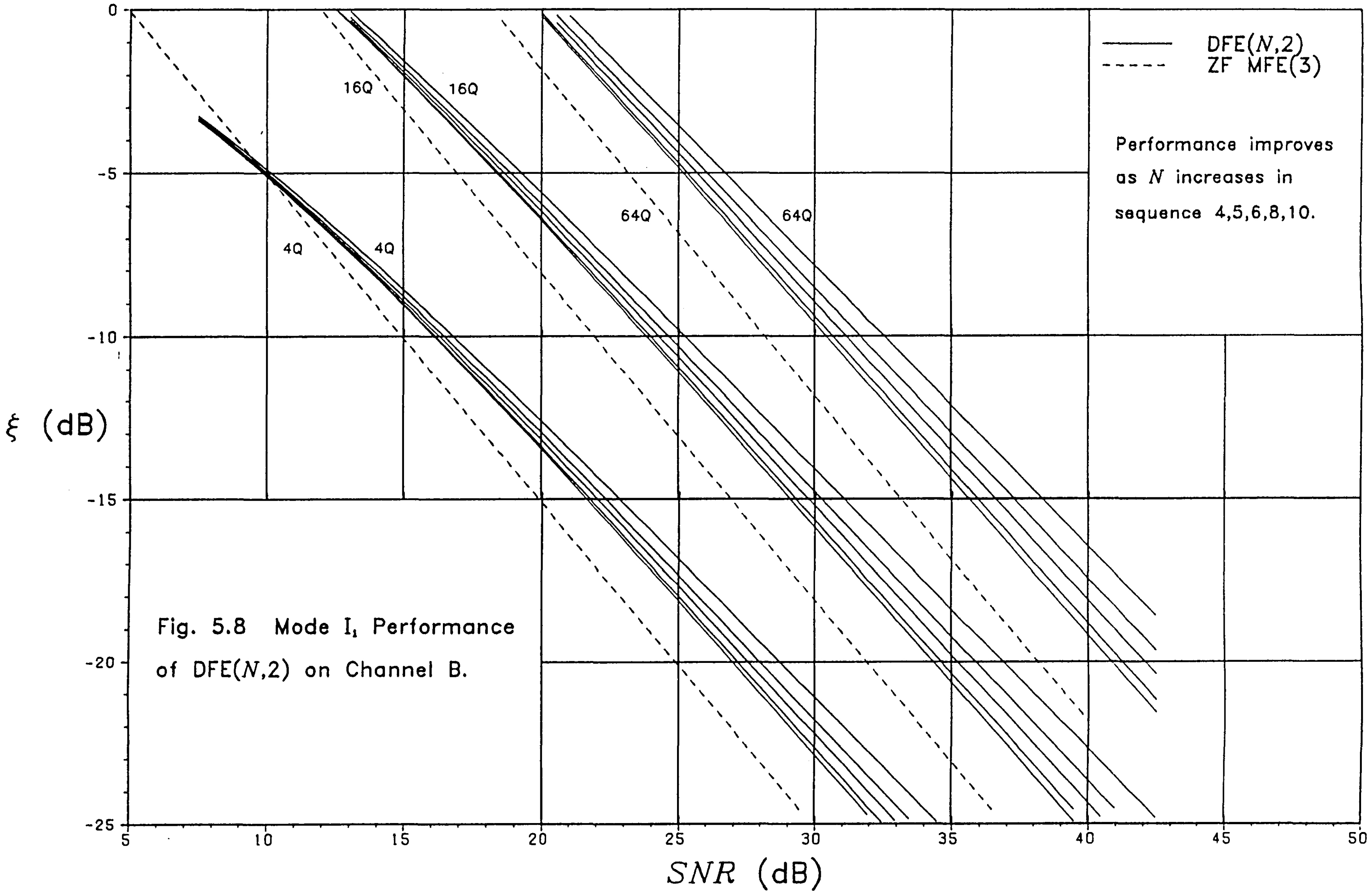


Fig. 5.7 Mode I_1 Performance of DFE(N,2) on Channel B.



Performance improves as N increases in sequence 4,5,6,8,10.

Fig. 5.8 Mode I_1 Performance of DFE(N,2) on Channel B.

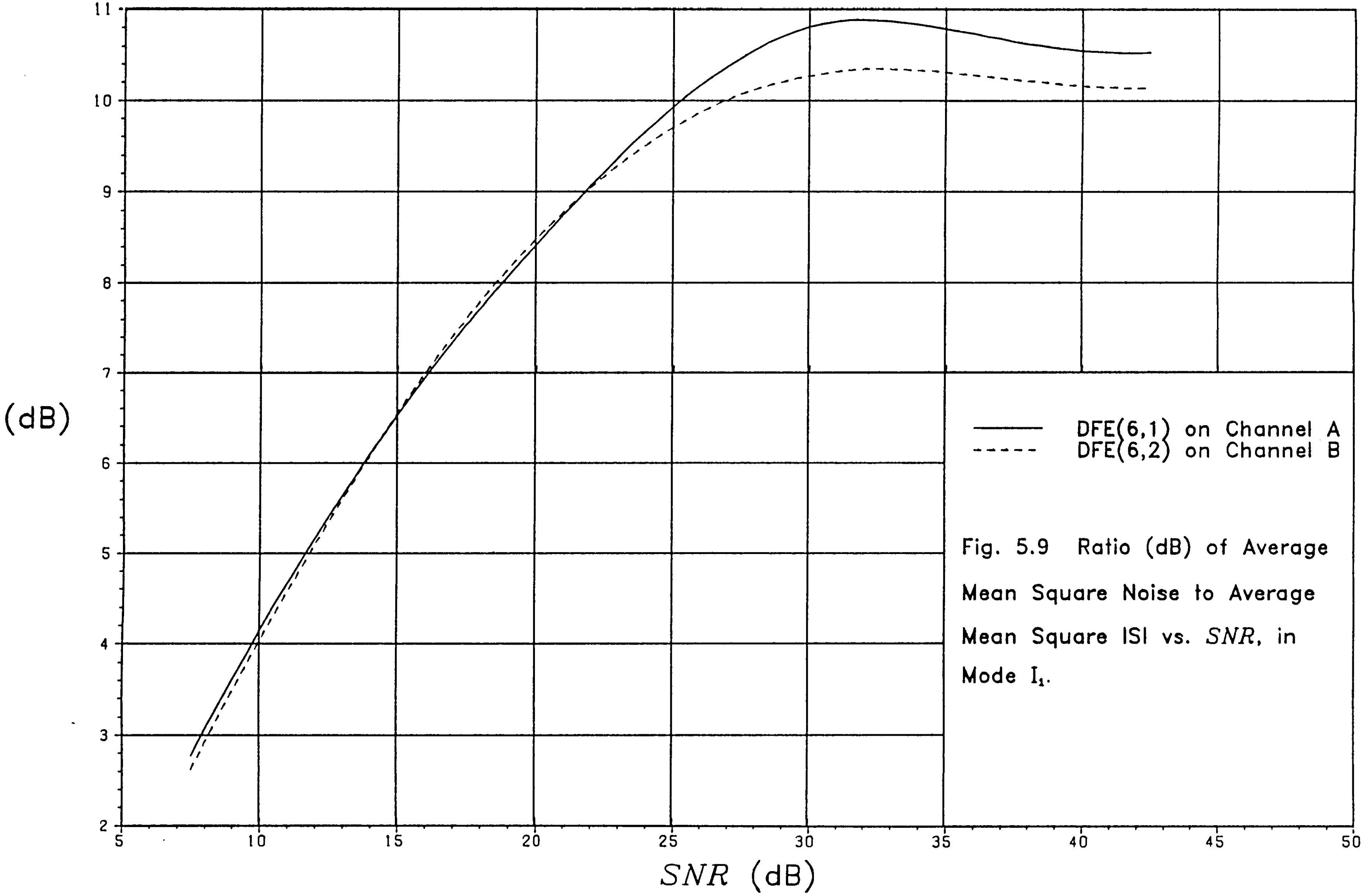


Fig. 5.9 Ratio (dB) of Average Mean Square Noise to Average Mean Square ISI vs. *SNR*, in Mode I_1.

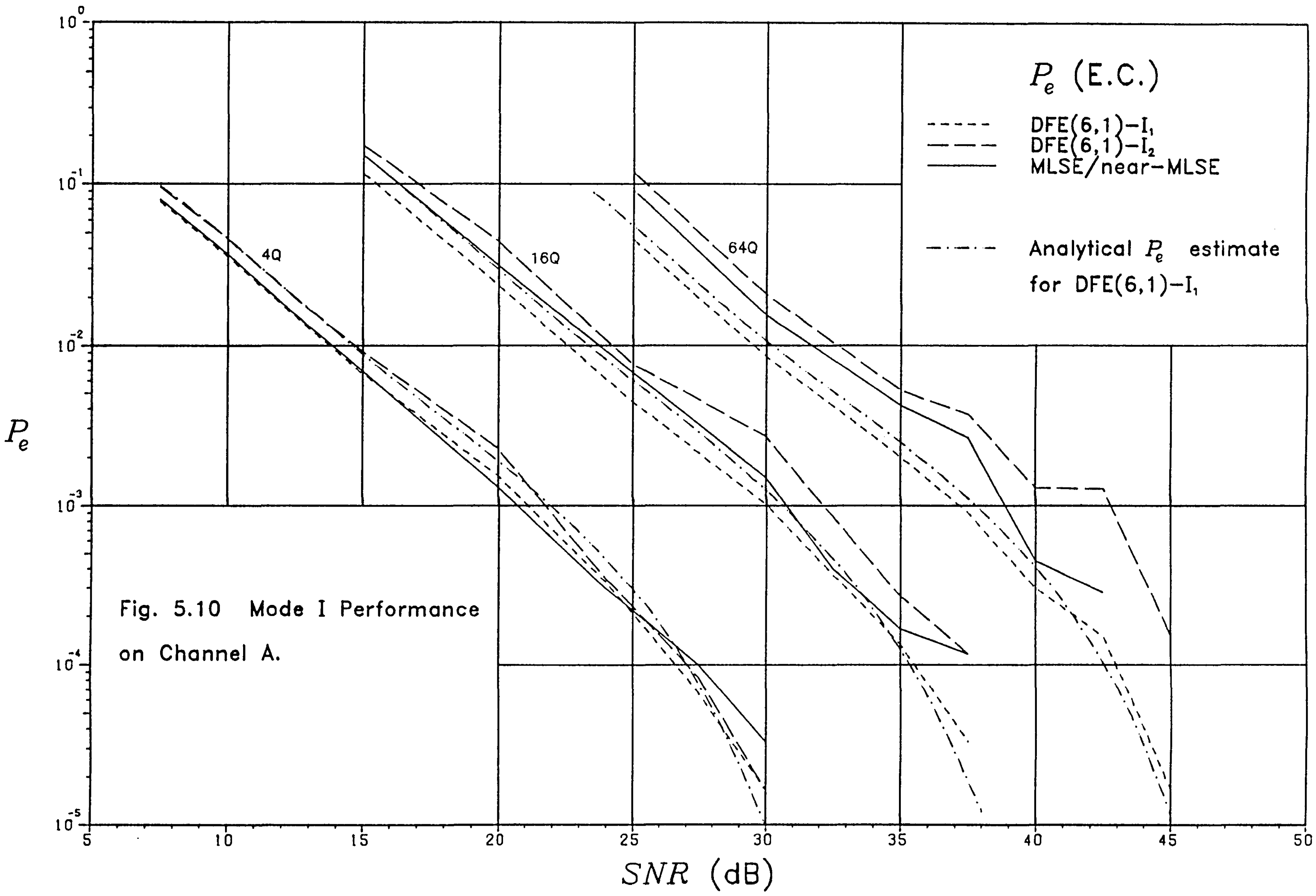


Fig. 5.10 Mode I Performance on Channel A.

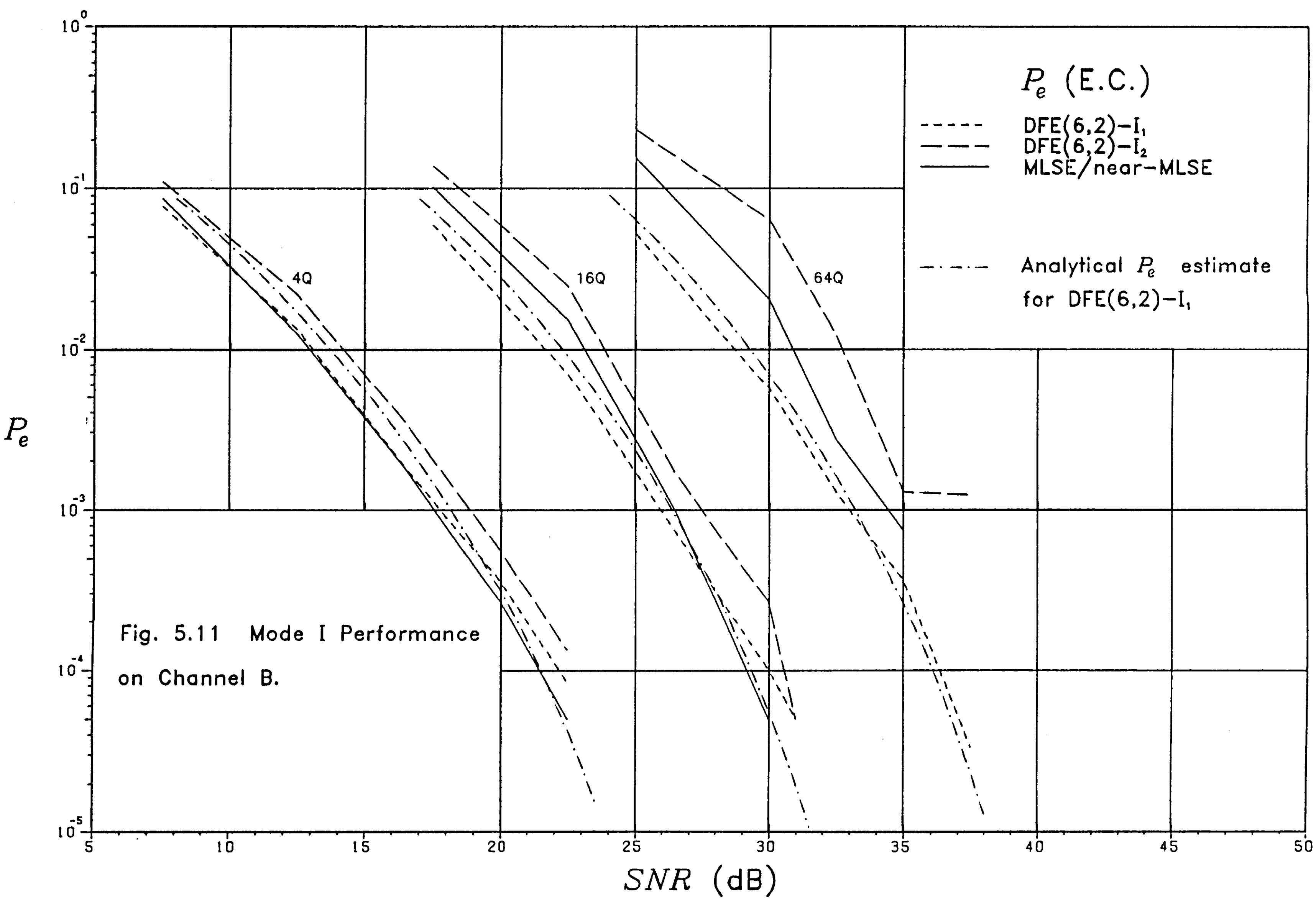


Fig. 5.11 Mode I Performance on Channel B.

Chapter 6
CHANNEL ESTIMATION

To implement the MLSE by the VA the receiver needs to know, and therefore track, the resultant channel impulse response vector $\underline{y}_k$. If a reduced-state VA is used, it is usual to have an adaptive linear filter ahead of the detection device to provide phase equalization, the resultant impulse response now being of minimum phase. This filter is ideally the feedforward section of a ZF DFE(∞,g) which in practice can be approximated, with low noise and N large enough, by the feedforward section of an MMSE DFE(N,g). Thus the impulse response vector seen by the detection device is assumed to be $[1\ b_{k1} \ldots b_{kg}]^{\mathrm{t}}$.

For a near-MLSE (requiring a pre-filter) or a DFE, the conventional approach to providing adaptation is to adjust the filter taps (both feedforward and feedback) directly, as explained in section 3.3.1. This procedure can be viewed as an indirect way of tracking the channel response vector $\underline{y}_k$, since it is theoretically possible (though perhaps tedious) to derive the channel vector from the correctly adjusted filter taps. Actually, a receiver using any detection process need only really track the channel vector $\underline{y}_k$ in order to be adaptive, since knowledge of this along with the data symbols allows us to fully describe the information bearing part of the received sample r_k, and so therefore all necessary information on the transmission medium is available in $\underline{y}_k$. It is shown in the next chapter that the performance of the DFE can be improved by using an alternative adaptation procedure to the conventional SRK one, basically that of tracking the channel vector directly and then computing the MMSE filter taps from it. This chapter is therefore devoted to the problem of channel estimation, comparing the performance of the SRK and SD algorithms. We will also attempt to derive theoretical expressions for performance and compare with experiment.

6.1 Channel Estimation using the RLS (SRK) Algorithm

To judge the worth of the SRK algorithm at various *SNR*'s, fade rates etc., we should observe the performance with the weighting coefficient ω optimized at each point. Rather than go through the laborious process of optimizing ω by measurement, it would be nice if one could derive the optimum value by analysis. The criterion of optimality we choose is the minimization of the average mean square error, ϵ, given by

$$\epsilon = \mathrm{E}[|\epsilon_k|^2] = \mathrm{E}[|r_k - \underline{\mathbf{x}}_k^{\mathrm{t}}\underline{\mathbf{c}}_{k-1}|^2] \tag{6.1}$$

where the expectation is over the data symbols, noise and fading statistics. Recall from section 3.2.2 that $\underline{\mathbf{x}}_k$, with detection delay q=0, is

$$\underline{\mathbf{x}}_k^{\mathrm{t}} = [\hat{s}_k \ \ldots \ \hat{s}_{k-g}] \tag{6.2}$$

where it will be assumed that the decisions $\{\hat{s}_k\}$ are correct. Substituting $r_k = \underline{\mathbf{x}}_k^{\mathrm{t}}\underline{\mathbf{y}}_k + n_k$, and neglecting the dependence between $\underline{\mathbf{x}}_k$ and $\underline{\mathbf{c}}_{k-1}$,

$$\epsilon = \mathrm{E}[(\underline{\mathbf{y}}_k - \underline{\mathbf{c}}_{k-1})^{*\mathrm{t}}\underline{\mathbf{x}}_k^*\underline{\mathbf{x}}_k^{\mathrm{t}}(\underline{\mathbf{y}}_k - \underline{\mathbf{c}}_{k-1})] + \sigma_n^2 = \sigma_s^2\mathrm{E}[\|\underline{\mathbf{y}}_k - \underline{\mathbf{c}}_{k-1}\|^2] + \sigma_n^2 \tag{6.3}$$

where we have used

$$\mathrm{E}[\underline{\mathbf{c}}_{k-1}n_k] = \underline{\mathbf{0}}$$

$$\mathrm{E}[\underline{\mathbf{x}}_k^*\underline{\mathbf{x}}_k^{\mathrm{t}}] = \sigma_s^2\mathrm{I} \tag{6.4}$$

Minimization of ϵ is achieved by minimizing, with respect to ω, the quantity ϵ' which we define as the error in the channel estimate, i.e.

$$\epsilon' = \mathrm{E}[\|\underline{\mathbf{y}}_k - \underline{\mathbf{c}}_{k-1}\|^2] \tag{6.5}$$

In [82], minimization of ϵ' for an RLS algorithm is done for a simple channel model (not HF) that obeys a first order Markov process, that is, $\underline{\mathbf{y}}_k$ is a function of $\underline{\mathbf{y}}_{k-1}$ only. The nature of the HF channel suggests that $\underline{\mathbf{y}}_k$ may effectively be a function of previous vectors as high as five intervals away [66], i.e. obey a fifth order Markov process.

Recall from chapter 3 that

$$\underline{\mathbf{c}}_{k-1} = \mathrm{R}_{x0}^{-1}(\sum_{i=0}^{k-1}\omega^{k-1-i}\underline{\mathbf{x}}_i^* r_i) \tag{6.6}$$

where

$$\mathrm{R}_{x0} = \sum_{i=0}^{k-1}\omega^{k-1-i}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}} \tag{6.7}$$

Let $$\epsilon'(p) = \mathrm{E}[\|\underline{\mathbf{y}}_p - \underline{\mathbf{c}}_{k-1}\|^2] \qquad \text{for} \quad p \geq k \tag{6.8}$$

Substituting (6.6) into (6.8),

$$\begin{aligned}\epsilon'(p) &= \mathrm{E}[\ \left\| \mathrm{R}_{x0}^{-1}\Big(\sum_{i=0}^{k-1}\omega^{k-1-i}\underline{\mathbf{x}}_i^*\{\underline{\mathbf{x}}_i^{\mathrm{t}}(\underline{\mathbf{y}}_p-\underline{\mathbf{y}}_i)-n_i\}\Big)\right\|^2\ ] \\ &= \mathrm{E}[\mathrm{Tr}\Big(\mathrm{R}_{x0}^{-1}\{(\sum_{i=0}^{k-1}\sum_{h=0}^{k-1}\omega^{k-1-i}\omega^{k-1-h}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}}\mathrm{E}[(\underline{\mathbf{y}}_p-\underline{\mathbf{y}}_i)(\underline{\mathbf{y}}_p-\underline{\mathbf{y}}_h)^{*\mathrm{t}}]\underline{\mathbf{x}}_h^*\underline{\mathbf{x}}_h^{\mathrm{t}}) \\ &\qquad\qquad + \sigma_n^2\sum_{i=0}^{k-1}\omega^{2(k-1-i)}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}}\}\mathrm{R}_{x0}^{-1}\Big)\ ]\end{aligned} \tag{6.9}$$

where Tr(.) means the trace of a matrix. From the assumed fading statistics of Channel A and Channel B (recall (4.21)), and remembering that $\mathrm{E}[y_{pi}y_{hm}^*]=0$ for $i \neq m$,

$$\begin{aligned}\mathrm{E}[(\underline{\mathbf{y}}_p-\underline{\mathbf{y}}_i)(\underline{\mathbf{y}}_p-\underline{\mathbf{y}}_h)^{*\mathrm{t}}] &= \mathrm{E}[\underline{\mathbf{y}}_p\underline{\mathbf{y}}_p^{*\mathrm{t}}-\underline{\mathbf{y}}_i\underline{\mathbf{y}}_p^{*\mathrm{t}}-\underline{\mathbf{y}}_p\underline{\mathbf{y}}_h^{*\mathrm{t}}+\underline{\mathbf{y}}_i\underline{\mathbf{y}}_h^{*\mathrm{t}}] \\ &= \frac{1}{(g+1)}(1-e^{-\beta(p-i)^2/2}-e^{-\beta(p-h)^2/2}+e^{-\beta(i-h)^2/2})\mathrm{I}\end{aligned} \tag{6.10}$$

where $$\beta = (2f_r\pi T)^2 \tag{6.11}$$

We shall assume in the exponential summation of (6.9) that the optimum setting of ω will be small enough so that effectively $\beta(p-h)^2$, $\beta(p-i)^2$ and $\beta(i-h)^2 \ll 1$ at all times. Thus we see that (6.10) is effectively

$$\begin{aligned}\mathrm{E}[(\underline{\mathbf{y}}_p-\underline{\mathbf{y}}_i)(\underline{\mathbf{y}}_p-\underline{\mathbf{y}}_h)^{*\mathrm{t}}] &\approx \frac{1}{(g+1)}(1-1+\beta(p-i)^2/2-1+\beta(p-h)^2/2+1-\beta(i-h)^2/2)\mathrm{I} \\ &= \frac{\beta}{(g+1)}(p-i)(p-h)\mathrm{I}\end{aligned} \tag{6.12}$$

and substituting this in (6.9) gives

$$\epsilon'(p) \approx \mathrm{E}[\mathrm{Tr}\Big(\mathrm{R}_{x0}^{-1}\{\frac{\beta}{(g+1)}\mathrm{R}_{x1}^2+\sigma_n^2\sum_{i=0}^{k-1}\omega^{2(k-1-i)}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}}\}\mathrm{R}_{x0}^{-1}\Big)] \tag{6.13}$$

where $$\mathbf{R}_{x1} = \sum_{i=0}^{k-1}(p-i)\omega^{k-1-i}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}} \tag{6.14}$$

with the expectation in (6.13) now being only over the $\{\underline{\mathbf{x}}_i\}$. Since we are interested in the error over the long-term, k shall be assumed to be very large. We will now make an assumption, commonly used in the analysis of RLS algorithms (e.g. [82]) to greatly simplify matters, that $\mathbf{R}_{x0}$ can be effectively approximated as

$$\mathbf{R}_{x0} = \sum_{i=0}^{k-1} \omega^{k-1-i} \underline{\mathbf{x}}_i^* \underline{\mathbf{x}}_i^{\mathrm{t}} \approx \sum_{i=0}^{k-1} \omega^{k-1-i} \mathrm{E}[\underline{\mathbf{x}}_i^* \underline{\mathbf{x}}_i^{\mathrm{t}}] = \frac{\sigma_s^2}{(1-\omega)} \mathbf{I} \tag{6.15}$$

and also that

$$\sum_{i=0}^{k-1} \omega^{2(k-1-i)} \underline{\mathbf{x}}_i^* \underline{\mathbf{x}}_i^{\mathrm{t}} \approx \frac{\sigma_s^2}{(1-\omega^2)} \mathbf{I} \tag{6.16}$$

Thus $\mathbf{R}_{x1}$ becomes

$$\begin{aligned} \mathbf{R}_{x1} &= \omega^{k-p} \frac{d}{d\omega} [\omega^{p-k+1} \mathbf{R}_{x0}] \\ &\approx \omega^{k-p} \sigma_s^2 \frac{d}{d\omega} [\frac{\omega^{p-k+1}}{(1-\omega)}] \mathbf{I} = \frac{\sigma_s^2}{(1-\omega)^2} (1+(p-k)(1-\omega)) \mathbf{I} \end{aligned} \tag{6.17}$$

The approximations of (6.15) and (6.16) should be more accurate the closer ω is to 1, since they then become more like long-term time averages. Using (6.15)–(6.17) in (6.13) we hence obtain

$$\epsilon'(p) = \frac{\beta}{(1-\omega)^2} (1+(p-k)(1-\omega))^2 + (g+1)\rho \frac{(1-\omega)}{(1+\omega)} \tag{6.18}$$

and

$$\epsilon' = \epsilon'(k) = \frac{\beta}{(1-\omega)^2} + (g+1)\rho \frac{(1-\omega)}{(1+\omega)} \tag{6.19}$$

$$\epsilon = \sigma_s^2 (\epsilon' + \rho) \tag{6.20}$$

The first term on the RHS of (6.19) is due to the time variation of the channel, and as such a quantity of this nature is usually referred to as the "lag" since it is the error caused by the algorithm's limitation in coping with time variations. The second term on the RHS of (6.19) is due to the additive gaussian noise, and is proportional to the length of the channel vector. Eq. (6.19) tells us, not surprisingly, that as ω decreases, the amount of lag-error reduces but the noise-error increases. An optimum setting of $\omega=1$ is achieved when $\beta=0$, i.e. no time variations, so that the lag-error is not infinite but zero.

The essential difference of (6.19) with the expression for a first-order Markov channel [82], is that the denominator of the lag term in the latter is only proportional to $(1-\omega)$.

Differentiating ϵ' with respect to ω,

$$\frac{d\epsilon'}{d\omega} = \frac{2\beta}{(1-\omega)^3} - \frac{2(g+1)\rho}{(1+\omega)^2} \tag{6.21}$$

and setting it to zero gives the optimum ω as the solution to

$$SNR = \frac{1}{\rho} = \frac{(g+1)}{\beta}\frac{(1-\omega)^3}{(1+\omega)^2} \tag{6.22}$$

SNR is plotted against optimum ω in fig. 6.1 for $2f_r=1$ and 2 Hz. The curves predict that above a certain *SNR* a positive optimum value of ω is not possible, indeed it will actually be negative. This result arises because of the approximations of (6.15) and (6.16), which were initially made under the assumption that $\omega>0$ (and preferably close to 1), but definitely not $\omega<0$. Note also that $\omega\neq0$ as then $\mathbf{R}_{x0}$ theoretically has no unique inverse, a situation which again (6.15) and (6.16) do not appear affected by. We should thus not be surprised if the low-ω behaviour predicted by our theoretical derivations is somewhat inconsistent with experiment. Substituting (6.22) into (6.19) gives

$$\epsilon'_{\min} = \beta\frac{(2+\omega)}{(1-\omega)^2} \tag{6.23}$$

where ω is at its optimum value. Figs. 6.2 and 6.3 show the measured values of ϵ/σ_s^2 and ϵ' (time averages of $|\epsilon_k|^2/\sigma_s^2$ and $\|\underline{y}_k-\underline{c}_{k-1}\|^2$ respectively) from using the SRK algorithm, with known data symbols and optimum ω from fig. 6.1, over the fading patterns for Channel A and Channel B respectively. Before actually commencing its run over the 25 fade period patterns, the SRK algorithm has been running for 1.67 fade periods already (equivalent to 4000 and 2000 symbol intervals for f.r.=1.0 Hz and 2.0 Hz respectively), ensuring that any start-up transient effects have died away. The parameter η only influences the start-up convergence of the algorithm, as was discussed in chapter 3, and later on we will show how. For the moment, however, it will suffice to say that within the range 0.001 to 10.0, η has no influence because of the more than sufficient warm-up time of 1.67 fade periods.

Also shown in figs. 6.2 and 6.3 are the theoretical values obtained from (6.19) and (6.20). The theoretical expressions indicate that ϵ/σ_s^2 and ϵ' should be independent of the constellation size L^2. This was found to be virtually so, the maximum deviation observed between 4, 16 and 64 QAM over all readings being about 0.1 dB. Thus the experimental curves in figs. 6.2 and 6.3 are effectively valid for 4, 16 and 64 QAM.

The agreement between theory and experiment over the range of *SNR* considered is reasonably good, the agreement worsening as the *SNR* increases, or rather as the "optimum" ω setting decreases. This, as already mentioned, is due to the approximations of (6.15) and (6.16) which we should expect to become less accurate as ω decreases. Also, the agreement for Channel A is slightly better than for Channel B. This can again be put down to the approximations of (6.15) and (6.16), since for larger g there is greater chance of there being some non-zero elements not on the main diagonal in the LHS of these equations, thus making them less like a diagonal matrix. A supposedly more accurate approximation to (6.15) is offered, without the derivation, in [82], putting

$$\sum_{i=0}^{k-1}\omega^{k-1-i}\underline{\mathrm{x}}_i^*\underline{\mathrm{x}}_i^{\mathrm{t}} \approx \frac{\sigma_s^2}{(1-\omega)}\left\{1+\sqrt{\frac{(1-\omega)}{(1+\omega)}\gamma}\right\}^{-1}\mathrm{I} \tag{6.24}$$

where $$\gamma = \frac{\mathrm{Var}[|x_i|^2]}{(\mathrm{Var}[x_i])^2} = \frac{\mathrm{E}[|s_i|^4]-(\mathrm{E}[|s_i|^2])^2}{(\mathrm{E}[|s_i|^2])^2} = \frac{2(L^2-4)}{5(L^2-1)} \tag{6.25}$$

A method of computation of $\mathrm{E}[|s_i|^4]$ can be found in the appendix of [92]. For L^2=4, 16 and 64 we have γ=0, 0.32 and 0.381 respectively, and γ is always less than 0.4. The modified approximation of (6.24) is identical to (6.15) for γ=0, and close to it for $\gamma \neq 0$ and ω close to 1. Our experimental measurements show that the dependence on γ is virtually negligible, so using (6.24), which complicates matters anyway, in place of (6.15) would not appear to help.

As the noise gets very low the lag-error is expected to dominate the channel error, and for no noise there should be an irreducible lower level representing the tracking algorithm's limit in coping with time variations. With ω set to its optimum value, the theoretical derivations give the ratio of lag-error to noise-error as $(1+\omega)^{-1}$, implying that for positive ω the noise is always bigger; we know already that at low ω this theoretical estimate of the ratio may be unreliable. The experimental curves in figs. 6.2 and 6.3 are actually starting to show a very slight inclination to "level out" at SNR=45.0 dB. We have measured the lag-to-noise error ratio at SNR=45.0 dB, 2.0 Hz fade rate, to be approximately 1.18 and 1.58 on Channel A and Channel B repectively, the corresponding theoretical predictions being 0.71 and 0.68 respectively. More will be said about the algorithm's limitation in coping with time variations later on.

At a given SNR, the channel error is lower on Channel A than on Channel B, which is due to the latter being represented by a longer channel vector.

Fig. 6.4 shows, for both theory and experiment, the variation of the channel error ϵ' with ω at SNR=24.0 dB and 45.0 dB on Channel A, and SNR=22.5 dB and 45.0 dB on Channel B, at a fade rate of 2.0 Hz. It can be seen that the agreement with theory is better for the lower SNR (with its higher range for ω) than for the higher SNR. Nevertheless, even at SNR=45.0 dB the measured ϵ' yielded by ω set to its theoretically optimum value (0.403 and 0.463 for Channel A and Channel B respectively) is very close to the measured ϵ'_{min}. The rate of change of ϵ' with respect to ω is smaller for the higher SNR, which is to be expected since for low noise ϵ' now depends to a greater degree on the comparatively less erratic time variations of the channel. Observe in fig. 6.4 that the difference in measured ϵ'_{min} between 4 QAM and 64 QAM at SNR=45.0 dB is no more than 0.1 dB, 4 QAM giving the lower value. In general the difference in ϵ' between the constellation sizes diminishes as ω increases, which we expect since the approximations of (6.15) and (6.16), which are independent of γ, become more accurate.

We will discuss the question of convergence from start-up after we have looked at the SD algorithm.

6.2 Channel Estimation using the SD Algorithm

It is well known that the SD algorithm performs a great deal worse than the RLS when used in the conventional tracking approach to the DFE, e.g. see [15], [64], [71]. This is because the covariance matrix $\mathbf{R}_k$ is highly dependent on the channel reponse, and its eigenvalue spread is thus constantly changing. It was pointed out in chapter 3 that a large eigenvalue spread leads to slower convergence and a more restricted range for stability on the parameter μ. Improving the performance by orthogonalization cannot be done because $\mathbf{R}_k$ is unknown.

The SD algorithm should ideally be used when $\mathbf{R}_k$ is known, so that orthogonalization can be achieved. This is automatically done in channel estimation because

$$\mathbf{R}_k = \sigma_s^2\mathbf{I} \tag{6.26}$$

$\mathbf{R}_k$ now being the covariance matrix of data symbols. Before comparing the performance of SD channel estimation with that of SRK, we will derive the optimum setting for the parameter μ. Recalling (3.15),

$$\begin{aligned} \underline{\mathbf{c}}_k &= \underline{\mathbf{c}}_{k-1} + \mu\underline{\mathbf{x}}_k^*\epsilon_k \\ &= (\mathbf{I} - \mu\underline{\mathbf{x}}_k^*\underline{\mathbf{x}}_k^{\mathrm{t}})\underline{\mathbf{c}}_{k-1} + \mu\underline{\mathbf{x}}_k^*(\underline{\mathbf{x}}_k^{\mathrm{t}}\underline{\mathbf{y}}_k + n_k) \\ &= \prod_{i=1}^{k}(\mathbf{I} - \mu\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}})\underline{\mathbf{c}}_0 + \sum_{i=1}^{k}\Big\{\prod_{m=i+1}^{k}(\mathbf{I} - \mu\underline{\mathbf{x}}_m^*\underline{\mathbf{x}}_m^{\mathrm{t}})\Big\}\mu\underline{\mathbf{x}}_i^*(\underline{\mathbf{x}}_i^{\mathrm{t}}\underline{\mathbf{y}}_i + n_i) \end{aligned} \tag{6.27}$$

Assuming μ is within the range $0<\mu<2/((g+1)\sigma_s^2)$ required for stable convergence, the first term in (6.27) goes to zero as $k \rightarrow \infty$. Actually this range, given in (3.25) of chapter 3, was derived by ignoring the fourth-order statistics of $\underline{\mathbf{x}}_i$. Since $\underline{\mathbf{x}}_i$ is a vector of data symbols we may easily re-write (3.23) for this case more precisely as

$$\mathrm{E}[(\underline{\mathbf{x}}_k^*\underline{\mathbf{x}}_k^{\mathrm{t}})^2] = \mathrm{E}[(\underline{\mathbf{x}}_k^{\mathrm{t}}\underline{\mathbf{x}}_k^*)\underline{\mathbf{x}}_k^*\underline{\mathbf{x}}_k^{\mathrm{t}}] = \sigma_s^4(g+1+\gamma)\mathbf{I} \tag{6.28}$$

where γ was defined in (6.25). The range for stable convergence then becomes

$$0 < \mu < \frac{2}{\sigma_s^2(g+1+\gamma)} \tag{6.29}$$

Since $\gamma<0.4$ its effect may not make much difference. In any case, assuming stable convergence, we get for k large

$$
\begin{aligned}
\underline{\mathrm{c}}_{k-1} &\approx \sum_{i=1}^{k-1}\Big\{\prod_{m=i+1}^{k-1}(\mathrm{I}-\mu\underline{\mathrm{x}}_m^*\underline{\mathrm{x}}_m^{\mathrm{t}})\Big\}\mu\underline{\mathrm{x}}_i^*(\underline{\mathrm{x}}_i^{\mathrm{t}}\underline{\mathrm{y}}_i+n_i) \\
&= \sum_{i=1}^{k-1}\Big\{\prod_{m=i+1}^{k-1}(\mathrm{I}-\mu\underline{\mathrm{x}}_m^*\underline{\mathrm{x}}_m^{\mathrm{t}})\Big\}\underline{\mathrm{y}}_i-\sum_{i=1}^{k-1}\Big\{\prod_{m=i}^{k-1}(\mathrm{I}-\mu\underline{\mathrm{x}}_m^*\underline{\mathrm{x}}_m^{\mathrm{t}})\Big\}\underline{\mathrm{y}}_i \\
&\qquad +\mu\sum_{i=1}^{k-1}\Big\{\prod_{m=i+1}^{k-1}(\mathrm{I}-\mu\underline{\mathrm{x}}_m^*\underline{\mathrm{x}}_m^{\mathrm{t}})\Big\}\underline{\mathrm{x}}_i^* n_i
\end{aligned}
\tag{6.30}
$$

We shall make the following simplifying assumptions.

$$
\left.
\begin{aligned}
\mathrm{E}[\underline{\mathrm{x}}_i^*\underline{\mathrm{x}}_m^{\mathrm{t}}] &= 0 \qquad \text{for } i\neq m \\
&= \sigma_s^2\mathrm{I} \qquad \text{for } i=m
\end{aligned}
\right\}
\tag{6.31}
$$

$$
\left.
\begin{aligned}
\mathrm{E}[\underline{\mathrm{x}}_i^*\underline{\mathrm{x}}_i^{\mathrm{t}}\underline{\mathrm{x}}_m^*\underline{\mathrm{x}}_m^{\mathrm{t}}] &= \sigma_s^4\mathrm{I} \qquad \text{for } i\neq m \\
&= (g+1+\gamma)\sigma_s^4\mathrm{I} \qquad \text{for } i=m
\end{aligned}
\right\}
\tag{6.32}
$$

Let
$$\mu_1 = 1-\mu\sigma_s^2 \tag{6.33}$$

$$\mu_2 = 1-2\mu\sigma_s^2+(g+1+\gamma)\mu^2\sigma_s^4 > 0 \tag{6.34}$$

where the stability requirement of (6.29) ensures $|\mu_1|$ and μ_2 are <1. Then

$$
\begin{aligned}
\epsilon'(p) &= \mathrm{E}[\|\underline{\mathrm{y}}_p-\underline{\mathrm{c}}_{k-1}\|^2] \\
&= 1-2(1-\mu_1)\sum_{i=1}^{k-1}\mu_1^{k-1-i}e^{-\beta(p-i)^2/2}+\chi+(g+1)\sigma_n^2\sigma_s^2\mu^2\sum_{i=1}^{k-1}\mu_2^{k-1-i}
\end{aligned}
\tag{6.35}
$$

where

$$
\begin{aligned}
\chi &= \sum_{i=1}^{k-1}\sum_{h=1}^{k-1}\Big\{\mu_1^{|i-h|}\mu_2^{k-1-\max\{i,h\}}-2\mu_1^{|i+1-h|}\mu_2^{k-1-\max\{i,h-1\}} \\
&\qquad +\mu_1^{|i-h|}\mu_2^{k-\max\{i,h\}}\Big\}e^{-\beta(i-h)^2/2} \\
&= \sum_{i=1}^{k-1}\sum_{h=1}^{k-1}\mu_1^{|i-h|}\mu_2^{k-1-\max\{i,h\}}e^{-\beta(i-h)^2/2} \\
&\qquad -2\sum_{i=1}^{k-1}\sum_{h=0}^{k-2}\mu_1^{|i-h|}\mu_2^{k-1-\max\{i,h\}}e^{-\beta(i-h-1)^2/2} \\
&\qquad\qquad +\sum_{i=0}^{k-2}\sum_{h=0}^{k-2}\mu_1^{|i-h|}\mu_2^{k-1-\max\{i,h\}}e^{-\beta(i-h)^2/2}
\end{aligned}
$$

$$
\chi = \sum_{i=1}^{k-1}\sum_{h=1}^{k-1}\mu_1^{|i-h|}\mu_2^{k-1-\max\{i,h\}}(2e^{-\beta(i-h)^2/2}-e^{-\beta(i-h-1)^2/2}-e^{-\beta(h-i-1)^2/2})
$$

$$-\sum_{i=0}^{k-2}\mu_1^{k-1-i}e^{-\beta(k-1-i)^2/2}-\sum_{h=1}^{k-1}\mu_1^{k-1-h}e^{-\beta(k-1-h)^2/2}$$
$$+2\sum_{i=1}^{k-1}\mu_1^{k-1-i}e^{-\beta(k-i)^2/2}+\Big\{\sum_{i=0}^{k-2}\mu_1^i\mu_2^{k-1-i}e^{-\beta i^2/2}$$
$$+\sum_{h=1}^{k-1}\mu_1^h\mu_2^{k-1-h}e^{-\beta h^2/2}-2\sum_{i=1}^{k-1}\mu_1^i\mu_2^{k-1-i}e^{-\beta(i-1)^2/2}\Big\} \tag{6.36}$$

The last term on the RHS of (6.36), in $\{.\}$ brackets, is zero for k large. As in the previous analysis we make the approximation $e^{-\beta i^2}\approx 1-\beta i^2$, to get, for k large,

$$\chi \approx \sum_{i=1}^{k-1}\sum_{h=1}^{k-1}\mu_1^{|i-h|}\mu_2^{k-1-\max\{i,h\}}\beta-2\sum_{i=1}^{k-1}\mu_1^{k-1-i}\beta(k-i-\tfrac{1}{2})+1 \tag{6.37}$$

The first term on the RHS of (6.37) is simplified as

$$\sum_{i=1}^{k-1}\sum_{h=1}^{k-1}\mu_1^{|i-h|}\mu_2^{k-1-\max\{i,h\}}\beta = \beta\sum_{i=1}^{k-1}\Big(\sum_{h=1}^{i-1}\mu_1^{i-h}\mu_2^{k-1-i}+\sum_{h=i}^{k-1}\mu_1^{h-i}\mu_2^{k-1-h}\Big)$$

$$= \beta\sum_{i=1}^{k-1}\Big(\mu_2^{k-1-i}\,\frac{\mu_1(1-\mu_1^{i-1})}{(1-\mu_1)}+\mu_1^{k-1-i}\sum_{h=i}^{k-1}\Big(\frac{\mu_2}{\mu_1}\Big)^{k-1-h}\Big)$$

$$= \beta\Big(\frac{\mu_1}{(1-\mu_1)(1-\mu_2)}-\sum_{i=1}^{k-1}\Big\{\frac{\mu_1^i\mu_2^{k-1-i}}{(1-\mu_1)}+\frac{\mu_1^{k-1-i}(1-(\mu_2/\mu_1)^{k-i})}{(1-\mu_2/\mu_1)}\Big\}\Big)$$

with the first term in the $\{.\}$ being zero,

$$= \beta\Big(\frac{\mu_1}{(1-\mu_1)(1-\mu_2)}+\frac{1}{(1-\mu_2/\mu_1)}\Big\{\frac{1}{(1-\mu_1)}-\frac{(\mu_2/\mu_1)}{(1-\mu_2)}\Big\}\Big)$$

$$= \frac{\beta(1+\mu_1)}{(1-\mu_1)(1-\mu_2)} \tag{6.38}$$

The second term on the RHS of (6.37) is simplified by applying (6.17), to give us

$$2\beta\sum_{i=1}^{k-1}(k-i-\tfrac{1}{2})\mu_1^{k-1-i} = \frac{2\beta}{(1-\mu_1)^2}-\frac{\beta}{(1-\mu_1)} = \frac{\beta(1+\mu_1)}{(1-\mu_1)^2} \tag{6.39}$$

Thus
$$\chi = \frac{\beta(1+\mu_1)}{(1-\mu_1)(1-\mu_2)}-\frac{\beta(1+\mu_1)}{(1-\mu_1)^2}+1 \tag{6.40}$$

Consider the second term on the RHS of (6.35),

$$2(1-\mu_1)\sum_{i=1}^{k-1}\mu_1^{k-1-i}e^{-\beta(p-i)^2/2} \approx 2-(1-\mu_1)\beta\sum_{i=1}^{k-1}(p-i)^2\mu_1^{k-1-i} \tag{6.41}$$

Now

$$\sum_{i=1}^{k-1}(p-i)^2\mu_1^{k-1-i} = \sum_{i=1}^{k-1}\left\{\mu_1^{k-p+1}(p-i)(p-i-1)\mu_1^{p-2-i}+\mu_1^{k-p}(p-i)\mu_1^{p-1-i}\right\} \tag{6.42}$$

Using (6.17),

$$\sum_{i=1}^{k-1}(p-i)\mu_1^{p-1-i} = \frac{\mu_1^{p-k}}{(1-\mu_1)^2}[1+(p-k)(1-\mu_1)] \tag{6.43}$$

$$\begin{aligned}\sum_{i=1}^{k-1}(p-i)(p-i-1)\mu_1^{p-2-i} &= \frac{d}{d\mu_1}(\sum_{i=1}^{k-1}(p-i)\mu_1^{p-1-i})\\ &= \frac{(p-k)}{(1-\mu_1)^2}\mu_1^{p-k-1}[1+(p-k)(1-\mu_1)]\\ &\quad+\frac{\mu_1^{p-k}}{(1-\mu_1)^3}[2+(p-k)(1-\mu_1)]\end{aligned} \tag{6.44}$$

Therefore, (6.42) becomes

$$\sum_{i=1}^{k-1}(p-i)^2\mu_1^{k-1-i} = \frac{(1+\mu_1)}{(1-\mu_1)^3}+\frac{(p-k)}{(1-\mu_1)^2}[2+(p-k)(1-\mu_1)] \tag{6.45}$$

Collecting together (6.41), (6.45) and (6.40), we finally obtain from (6.35) that

$$\begin{aligned}\epsilon'(p) &= -1+\beta\frac{(1+\mu_1)}{(1-\mu_1)^2}+\beta\frac{(p-k)}{(1-\mu_1)}[2+(p-k)(1-\mu_1)]+\frac{\beta(1+\mu_1)}{(1-\mu_1)(1-\mu_2)}\\ &\quad-\beta\frac{(1+\mu_1)}{(1-\mu_1)^2}+1+(g+1)\rho\frac{(1-\mu_1)^2}{(1-\mu_2)}\end{aligned}$$

$$= \frac{\beta(1+\mu_1)}{(1-\mu_1)(1-\mu_2)}+\beta\frac{(p-k)}{(1-\mu_1)}[2+(p-k)(1-\mu_1)]+(g+1)\rho\frac{(1-\mu_1)^2}{(1-\mu_2)} \tag{6.46}$$

Substituting for μ_1 and μ_2 from (6.33) and (6.34), and letting $\mu'=\mu\sigma_s^2$, we obtain for the channel estimate error ϵ'

$$\begin{aligned}\epsilon' = \epsilon'(k) &= \mathrm{E}[\|\underline{\mathbf{y}}_k-\underline{\mathbf{c}}_{k-1}\|^2]\\ &= (2-\mu'(g+1+\gamma))^{-1}\left\{\beta\frac{(2-\mu')}{\mu'^2}+(g+1)\rho\mu'\right\}\end{aligned} \tag{6.47}$$

and from (6.3) the average algorithm error $\epsilon=\mathrm{E}[|\epsilon_k|^2]$ is,

$$\epsilon = \sigma_s^2(\epsilon'+\rho) \tag{6.48}$$

The first term on the expanded RHS of (6.47) is the lag-error, and the second term the noise-error (which is proportional to the length of the channel vector). Eq. (6.19) for the RLS algorithm and eq. (6.47) for the SD algorithm become identical when $g=0$, $\gamma=0$ and $\mu'=1-\omega$. In [82] it is shown that for a first-order Markov channel, the channel error for the RLS algorithm is identical to that for the SD algorithm. Setting $d\epsilon'/d\mu'=0$ gives us the optimum μ' as the solution of

$$SNR = \frac{1}{\rho} = \frac{(g+1)}{\beta}\frac{\mu'^3}{[4-\mu'-\mu'(3-\mu')(g+1+\gamma)]} \tag{6.49}$$

SNR is plotted against optimum μ' in figs. 6.5 and 6.6 for $\gamma=0$ and $\gamma=0.381$, corresponding to 4 QAM and 64 QAM respectively. The curve for $\gamma=0.32$, corresponding to 16 QAM, would lie in between the curves for $\gamma=0$ and $\gamma=0.381$, being closer to $\gamma=0.381$. The influence of γ appears to increase with increasing *SNR*. Notice that the optimum μ' is upper bounded by the smaller of the two positive roots of the quadratic expression in μ' in the denominator of (6.49), i.e.

$$\mu'_{\text{opt}} < 1.5\frac{\left(g+\frac{4}{3}+\gamma-\sqrt{(g+\gamma)(g+\gamma+\frac{8}{9})}\right)}{(g+1+\gamma)} \tag{6.50}$$

The upper bound of (6.50) is tighter than the stability bound of (6.29), a fact that can be verified by substituting $\mu'=2(g+1+\gamma)^{-1}$ in the denominator of (6.49), which gives a negative value, thus implying that this value of μ' lies in between the two roots of the quadratic. For Channel A the upper bound of (6.50) is approximately 0.719 and 0.595 for 4 and 64 QAM respectively, while for Channel B it is approximately 0.465 and 0.410 for 4 and 64 QAM respectively. As the fade rate increases, with a given constellation size, the optimum μ' needs to be larger, the amount by which becoming insignificant at very high *SNR* when for all fade rates the optimum μ' is close to the upper limit of (6.50).

In figs. 6.7 and 6.8 we have plotted the channel error ϵ' against *SNR*, with μ' set at its theoretically optimum value. The continuous dark curves are for the theoretical values of ϵ'

(6.47) while the plot-points are for the measured ϵ' over the fading patterns using the SD algorithm with known data symbols (and also a warm-up time of 1.67 fade periods to remove start-up transients). The agreement between theory and experiment is quite good, being better than that for the SRK algorithm at high *SNR*. The theoretical prediction of the lag-to-noise error ratio, with μ' set at its optimum value, is

$$\frac{\text{lag}}{\text{noise}} = \frac{\beta(2-\mu')}{\mu'^3(g+1)\rho} = \frac{(2-\mu')}{[4-\mu'-\mu'(3-\mu')(g+1+\gamma)]} \tag{6.51}$$

For μ' close to zero the ratio is about 0.5, indicating that the noise is twice as great as the lag (as it is for the SRK algorithm as well, with ω close to 1), and for μ' close to the upper bound of (6.50) the ratio becomes infinite, indicating that the lag is dominant. The theoretical curves in figs. 6.7 and 6.8 show the channel error gently starting to level-out, as confirmed by the experimental measurements. The theoretical lower limit of the channel error can be found by substituting the upper bound of (6.50) into the lag-error term of (6.47). We observe that the theoretical predictions of the SD algorithm's behaviour are far more consistent with what we expect, at high *SNR*, than are those derived for the SRK algorithm, probably because the assumptions made in the analysis for the simpler SD are less drastic.

Also shown in figs. 6.7 and 6.8 are the experimental ϵ' curves for the SRK algorithm, as shown previously in figs. 6.2 and 6.3. It can be seen that the SD algorithm performs at a comparable level to the SRK over virtually all of the range of *SNR* shown, the SD being slightly worse at the high *SNR* end, with this degradation being more for 64 QAM than for 4 QAM. At *SNR*=45.0 dB the difference between the measured ϵ' for the SD and SRK algorithms is at most about 1.5 dB, this being on Channel B at a 2.0 Hz fade rate with 64 QAM. It is also at this "setting" that the maximum difference in measured ϵ' occurs between 4 and 64 QAM for the SD algorithm, this being approximately 0.75 dB, compared with 0.1 dB for the SRK algorithm. It would appear that with increasing *SNR*, the constellation size, as characterized by the value of γ, is more influential in the SD than in the SRK algorithm, though the effect of this influence may still be considered negligible even at *SNR*=45.0 dB.

In fig. 6.9 we have plotted, for both theory and experiment, the variation of ϵ' with respect to μ', at SNR=24.0 dB and 45.0 dB on Channel A, and SNR=22.5 dB and 45.0 dB on Channel B, at a fade rate of 2.0 Hz. The agreement with theory is generally better at the lower SNR, and, as for the SRK algorithm, the rate of change of ϵ' with respect to the adaptation parameter, in this case μ', is smaller at the higher SNR. The prediction of the optimum setting for μ' is excellent.

6.3 Start-up Convergence of SRK and SD Algorithms

Figs. 6.10 and 6.11 show the start-up convergence of the error in the channel estimate, $\|\underline{y}_k-\underline{c}_{k-1}\|^2$, when the SRK and SD algorithms are used assuming known data symbols. At time 0 the estimated vector $\underline{c}_{-1}$ has all zero elements, the input vector $\underline{x}_0$ consists of data symbol values, and the starting conditions in the SRK algorithm are as given in Appendix C. Each of the curves in figs. 6.10 and 6.11 are the average of 30 separate runs, each run having a different sequence of gaussian noise samples and data symbols, but with the same channel variation over the 60 baud interval time span. For Channel A the channel vector varies from $(-0.641-j0.223\ ,\ -0.396+j0.354)$ to $(-0.666-j0.242\ ,\ -0.322+j0.502)$ over the 60 baud intervals, while on Channel B the variation is from $(-0.524-j0.182\ ,\ -0.324+j0.289\ ,\ -0.635-j0.064)$ to $(-0.544-j0.197\ ,\ -0.263+j0.410\ ,\ -0.628-j0.098)$.

For the SRK algorithm the parameter ω is set to its theoretically optimum value as given in fig. 6.1, while the parameter η is allowed to vary from $10^{-3}\sigma_s^2$ to $10.0\sigma_s^2$. It can be seen from the figures that the curves for all the different η's, for a given constellation size, do eventually come together and become indistinguishable from one another, verifying that the influence of η does decrease with time. A value of $\eta=10^{-2}\sigma_s^2$ appears to give a convergence rate close to optimum.

The convergence curves for the SD algorithm are obtained under exactly the same conditions as the SRK algorithm, with the parameter μ' set to its theoretically optimum value as given in figs. 6.5 and 6.6. Table 6.1 compares the convergence rates of the two algorithms, the values listed being those number of whole baud intervals needed for a curve to reach within 1.0 dB of its approximate "steady-state" value. For the SRK algorithm, the values in the table refer to the curves with $\eta = 10^{-2}\sigma_s^2$.

Channel conditions (Constellation size, channel, *SNR* (dB))	Time, in baud intervals, to reach specified dB value of channel error from start-up.	
	SRK algorithm	SD algorithm
4, A, 24.0	7 for −30 dB	42 for −30 dB
64, A, 45.0 4, A, 45.0	4 for −41 dB 5 for −41 dB	16 for −40 dB 10 for −41 dB
4, B, 22.5	9 for −27 dB	38 for −26 dB
64, B, 45.0 4, B, 45.0	4 for −40 dB 5 for −40 dB	30 for −39 dB 22 for −40 dB

Table 6.1 Comparison of convergence times for SRK and SD algorithms.

Table 6.1 clearly highlights the superiority of the SRK algorithm in reaching steady-state more rapidly, being about 2–6 times faster than the SD algorithm. The convergence of the SD algorithm can be improved, though perhaps not spectacularly so, by using a kind of "gear shifting" scheme in which the value of μ' is high at the beginning to provide rapid initial convergence, and then progressively reduced in size when the error is down to a certain level, e.g. see [57]. A much better method, though, is to use special periodic training sequences that enable a simple and close-to-steady-state calculation of the estimated vector to be made in baud intervals equal to the period (assuming the period is greater than the vector length), after which a tracking algorithm can take over. Convergence can thus be achieved in as fast a time as the RLS algorithm, and the training scheme is such that no symbol-synchronization is required at the receiver. In other words, the receiver does not need to know what the actual data symbols are at any particular instant, but only the cyclically ordered sequence in which they are transmitted. For further details on the use of periodic training sequences see [83], and in particular [84], which deals specifically with channel estimation.

Observe in Table 6.1 that at SNR=45.0 dB, the convergence of the SD algorithm is noticeably quicker with 4 QAM than with 64 QAM. This is due to the influence of γ, and not the different μ''s used, since a run with 64 QAM using the optimum μ' for 4 QAM, which is not shown, yielded little difference from the original. Contrast this behaviour with that of the SRK algorithm, which shows little difference between the constellation sizes.

One final interesting point is the slower convergence exhibited by the algorithms at the lower SNR values. We have mentioned how the long-term channel error is composed of two parts, namely the lag-error, which arises because of time variations of the channel, and the noise-error, which is due to additive gaussian noise. In order for the channel error to reach a steady-state both these components have to individually reach their respective steady-states. Usually at the start it is the lag-error which dominates the transient behaviour, but because the gaussian noise is, in the short-term, much more random in nature than the time variations of the channel, it is more likely the noise-error that takes longer to reach a steady-state, and this would be more so if the effective "window" size of the algorithm is larger. (A larger "window" means that the estimated vector at any instant depends to a greater extent in both degree and number on its past values, and this condition is brought about by a larger ω in the SRK algorithm or smaller μ' in the SD algorithm). Obviously for a lower SNR, the transient behaviour of the noise-error becomes more influential [93]. In particular, observe in figs. 6.10 and 6.11 the convergence curves for the SRK algorithm at the lower SNR with η set at its lower values. The curves have a sharp initial fall, but then decrease in slope considerably around a channel error value $\sim\rho$, and it is this change that makes convergence take longer than what it does at the higher SNR. In chapter 3 we discussed how the RLS algorithm achieves convergence in baud intervals of the order of the length of the estimated vector, this being independent of ω. That discussion can now be seen to be with relevance, strictly speaking, to the convergence of the lag-error, whose transient behaviour is responsible for the sharp initial fall in the convergence curves. For the SD algorithm, a smaller μ' also brings about slower convergence of the lag-error as well as the noise-error.

6.4 Improving the Performance at High *SNR*

The curves in figs. 6.7 and 6.8 indicate a very slight tendency for the channel error to start levelling out at high *SNR*, this being more apparent at the higher fade rate of 2.0 Hz. This is due to the lag-error starting to approach an irreducible level, this level representing the fundamental lower limit (in the absence of noise) of the algorithm's ability to track the channel's time variations.

The estimate of the channel vector at any instant is derived from previous received samples and data symbols, these being appropriately weighted according to their distance in the past. This can be most clearly seen in the RLS algorithm, where $\underline{c}_k$ is chosen such that it minimizes a weighted sum of time-spaced squared errors (3.33). The channel estimate is therefore a kind of "average vector" produced from previous samples and symbols, the effective time-span or window over which the average is formed being determined by the weighting. At low *SNR* the window has to be made large to limit the effect of noise. As the *SNR* increases the window can be made smaller, thereby reducing the lag-error because the channel estimate is now formed from a proportionally higher amount of more recent information. For any window setting, however, the estimation process does not take into account the variation in time of the channel vector over the past samples, and this is the fundamental limitation in the algorithm's tracking ability. The effect of this inadequacy is too small to make much difference when the noise level is sufficiently high, the effect becoming apparent only when the noise level has dropped enough to expose it. Hence the levelling-out of the channel error at high *SNR*. To improve matters beyond this point, then, it would seem necessary that the channel vector be estimated as a first or higher order function of time, ideally matching the short-term time variation of the channel. This would naturally entail an increase in the number of computations per update of the tracking process.

The expected value of the squared difference between two time separated values of the i^{th}

element of the channel vector is , recalling (4.21),

$$\mathrm{E}[|y_{ki}-y_{hi}|^2] = \mathrm{E}[|y_{ki}|^2-y_{ki}y_{hi}^*-y_{hi}y_{ki}^*+|y_{hi}|^2]$$

$$= \frac{2}{(g+1)}(1-e^{-\beta(k-h)^2/2}) \tag{6.52}$$

In the short-term we may assume $\beta(k-h)^2 \ll 1$. Therefore

$$\sqrt{\mathrm{E}[|y_{ki}-y_{hi}|^2]} \approx \sqrt{\frac{2}{(g+1)}(1-1+\frac{\beta(k-h)^2}{2})} = \sqrt{\frac{\beta}{(g+1)}}\;|k-h| \tag{6.53}$$

Eq. (6.53) suggests that, in the short-term, the channel variation is approximately linear with time. Therefore, estimating the channel vector as a linear function of time could be most appropriate. For example, we could determine at time kT the vectors $\underline{\mathbf{c}}_k$ and $\Delta\underline{\mathbf{c}}_k$ that minimize the weighted time average

$$\bar{\xi}_k = \sum_{i=0}^{k}\omega^{k-i}|r_i-\underline{\mathbf{x}}_i^{\mathrm{t}}\{\underline{\mathbf{c}}_k+\Delta\underline{\mathbf{c}}_k(i-k)\}|^2 \tag{6.54}$$

Vector $\Delta\underline{\mathbf{c}}_k$ is akin to a gradient vector, so that $\underline{\mathbf{c}}_k+\Delta\underline{\mathbf{c}}_k(i-k)$ is a kind of least squares linear estimate of the channel vector. The estimate of the channel vector $\underline{\mathbf{y}}_k$ for use at time kT would therefore be $\underline{\mathbf{c}}_{k-1}+\Delta\underline{\mathbf{c}}_{k-1}$. Differentiating (6.54) with respect to $\underline{\mathbf{c}}_k$ and setting the result to zero gives us

$$\underline{\mathbf{c}}_k = \Big(\sum_{i=0}^{k}\omega^{k-i}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}}\Big)^{-1}\Big\{\sum_{i=0}^{k}\omega^{k-i}\underline{\mathbf{x}}_i^*r_i+\Big(\sum_{i=0}^{k}(k-i)\omega^{k-i}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}}\Big)\Delta\underline{\mathbf{c}}_k\Big\} \tag{6.55}$$

With $\Delta\underline{\mathbf{c}}_k=\underline{\mathbf{0}}$, (6.55) is identical to what we originally had in (6.6). Differentiating (6.54) with respect to $\Delta\underline{\mathbf{c}}_k$ and setting the result to zero gives us

$$\Delta\underline{\mathbf{c}}_k = -\Big(\sum_{i=0}^{k}(k-i)^2\omega^{k-i}\underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^{\mathrm{t}}\Big)^{-1}\Big\{\sum_{i=0}^{k}(k-i)\omega^{k-i}\underline{\mathbf{x}}_i^*(r_i-\underline{\mathbf{x}}_i^{\mathrm{t}}\underline{\mathbf{c}}_k)\Big\} \tag{6.56}$$

Unfortunately the implementation of (6.55) and (6.56) as a recursive algorithm is considerably more complicated than that for $\Delta\underline{\mathbf{c}}_k=\underline{\mathbf{0}}$. Therefore more simpler schemes, still based on the notion of a channel estimate that is a linear function of time, are desired, and one such is given in [66]. This is based on an SD algorithm, though there is no reason to suggest this could not be replaced by an SRK algorithm if desired, that utilizes a weighted least squares linear fit of previous channel estimates. The primary objective of [66], however, was to see if a good estimate of the channel vector could be produced via the decisions of an MLSE, this device having a significant delay in detection. The estimation of the channel vector as a linear function of time allows us to predict, by extrapolation, what the vector could be in say 10–20 symbol intervals time. This is what is done in [66], and the results given suggest that this process of linear extrapolation can be quite effective in combatting the detection delay of an MLSE.

The form of the channel error curves in figs. 6.7 and 6.8 does not suggest any great degradation (up to SNR=45.0 dB) caused by an irreducible lag-error yet, although this could be different for longer channel vectors and/or higher fade rates. Furthermore, in our investigations we shall be ignoring, for purposes of channel estimation, the delay in detection of the MLSE/near-MLSE, the reason for which will be given in chapter 7. Therefore we shall not be implementing any modified versions of the SRK or SD algorithms that estimate the channel vector as a linear function of time.

On a passing note we mention a more different estimation algorithm developed in [85], which uses SD-type updating (therefore not too complex in computations) and which gives good performance by assuming knowledge of the actual configuration of the transmission medium, i.e. the number of paths present together with their relative strengths and time delays.

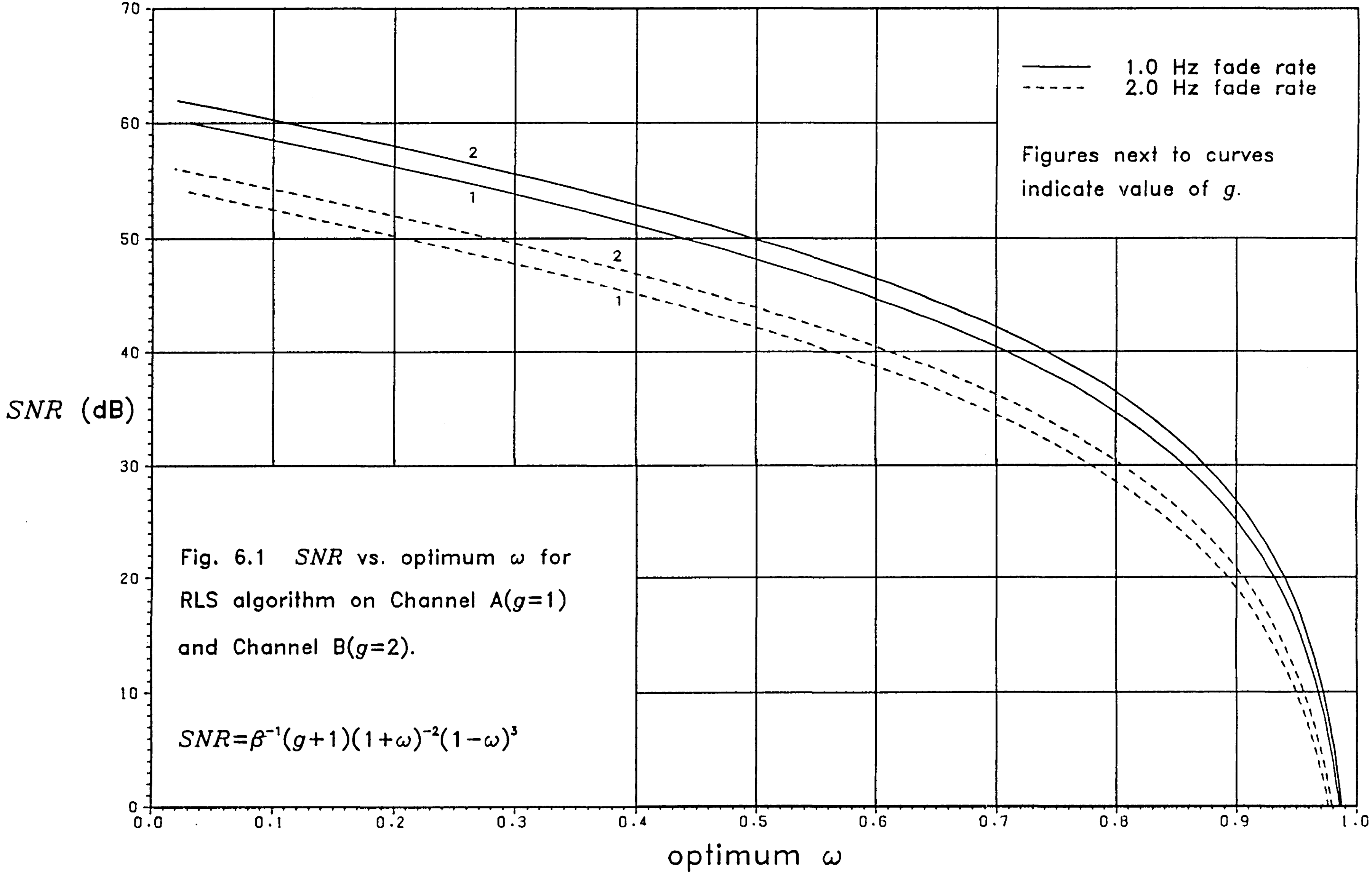


Fig. 6.1 *SNR* vs. optimum ω for RLS algorithm on Channel A(g=1) and Channel B(g=2).

$SNR=\beta^{-1}(g+1)(1+\omega)^{-2}(1-\omega)^{3}$

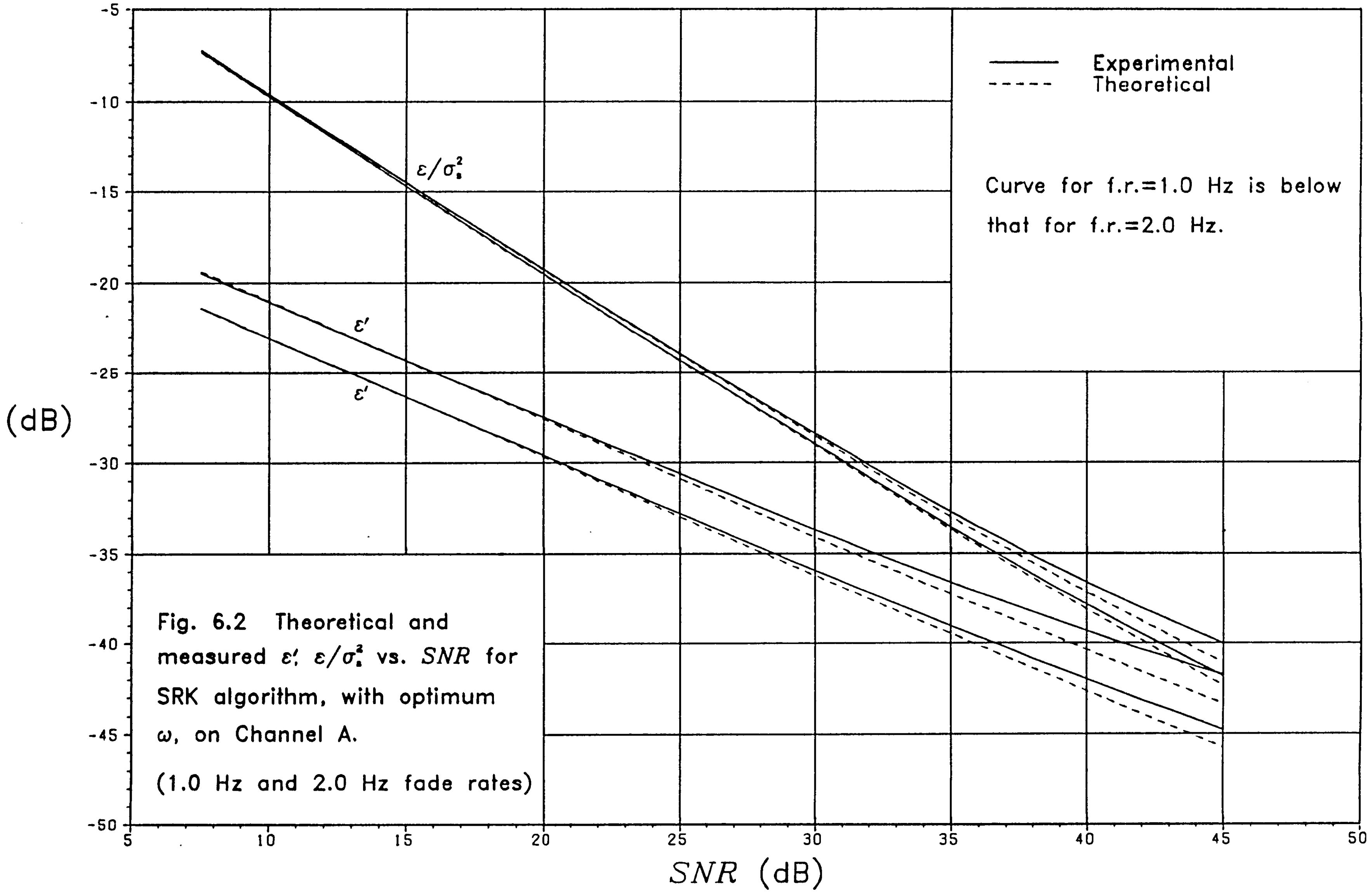
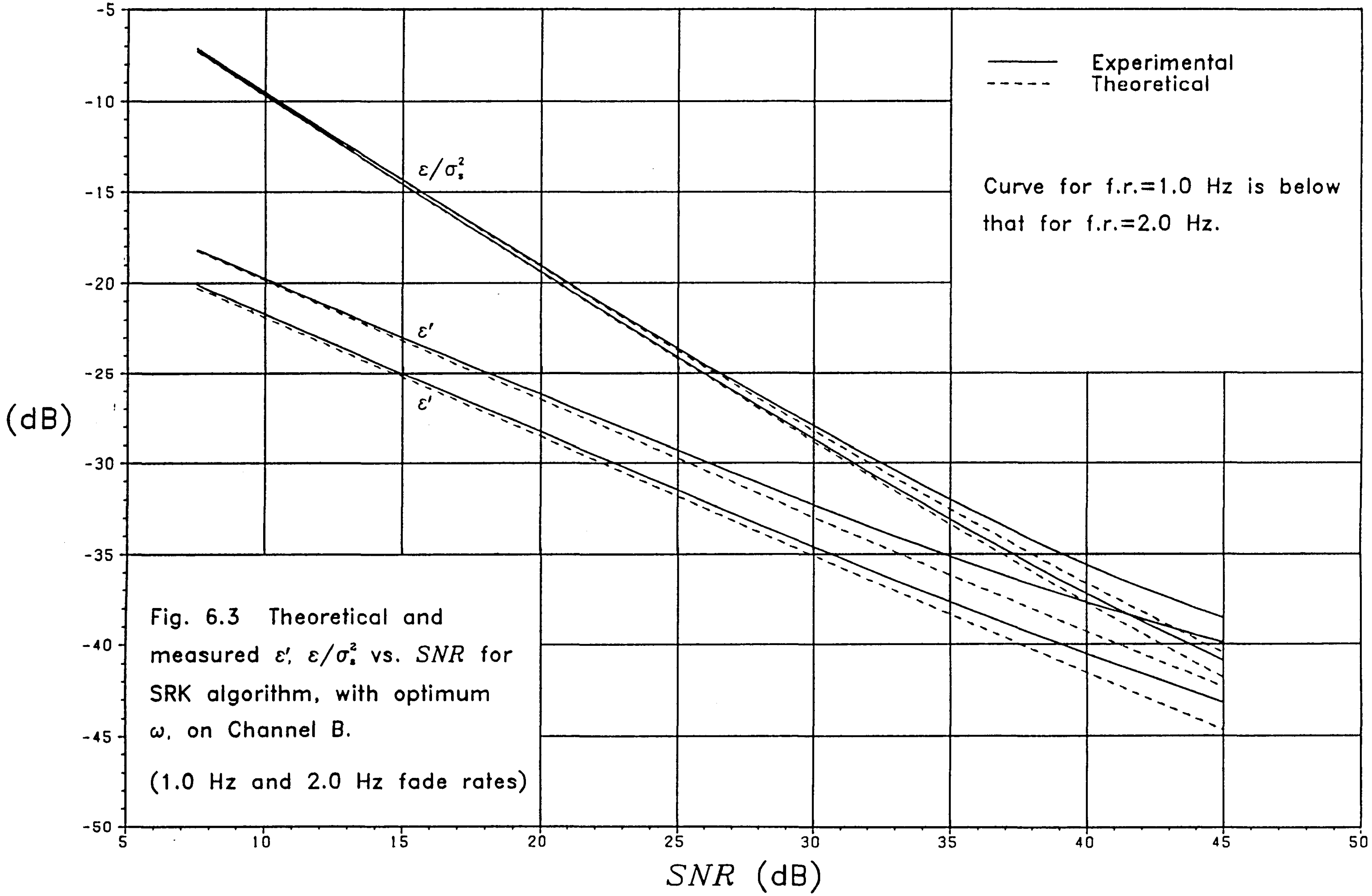


Fig. 6.2 Theoretical and measured ε', ε/σ_s^2 vs. SNR for SRK algorithm, with optimum ω, on Channel A.

(1.0 Hz and 2.0 Hz fade rates)



Curve for f.r.=1.0 Hz is below that for f.r.=2.0 Hz.

Fig. 6.3 Theoretical and measured ε', ε/σ_s^2 vs. SNR for SRK algorithm, with optimum ω, on Channel B.

(1.0 Hz and 2.0 Hz fade rates)

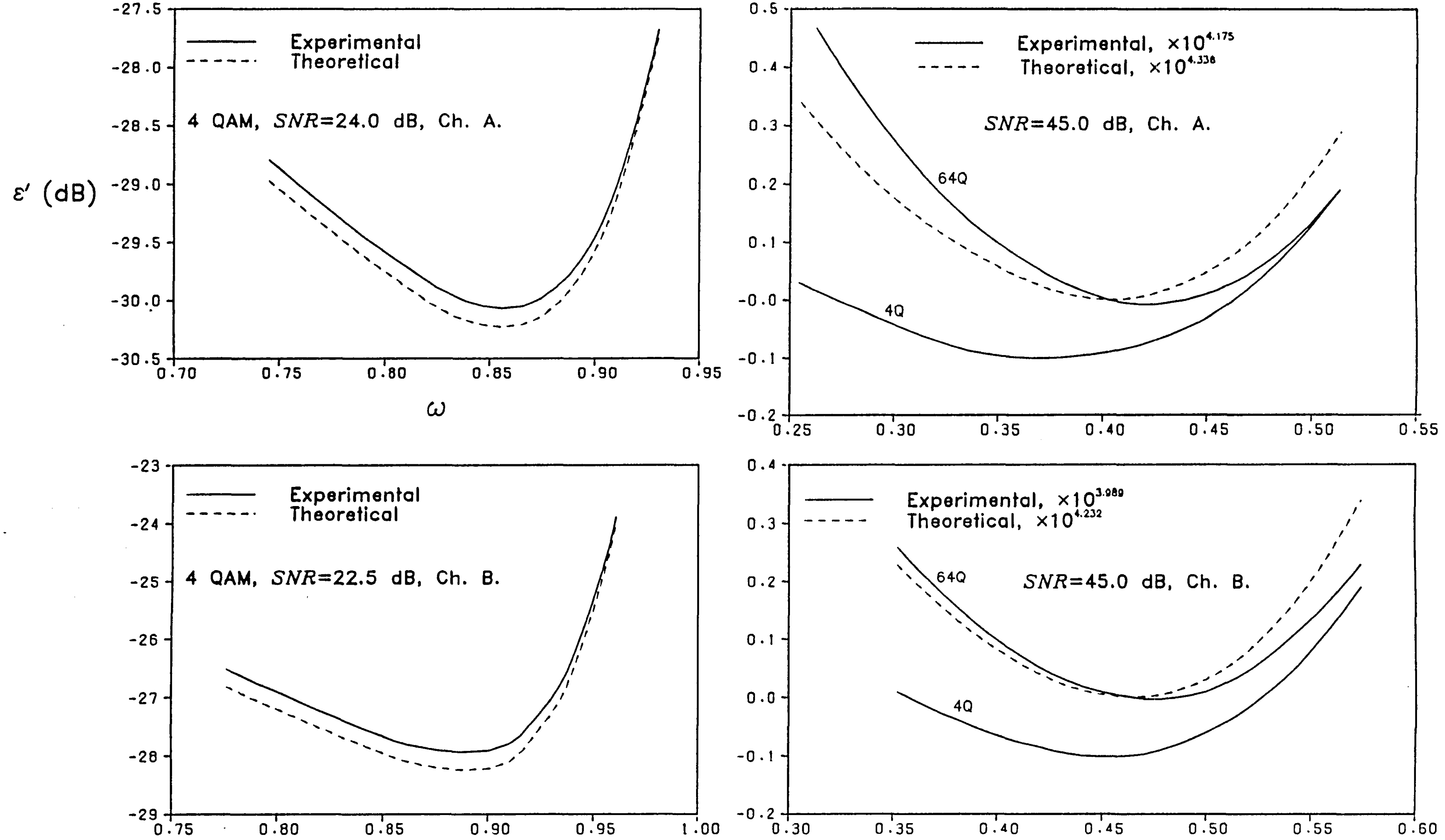


Fig. 6.4 Variation of ε′ with ω for SRK algorithm, f.r.=2.0 Hz.

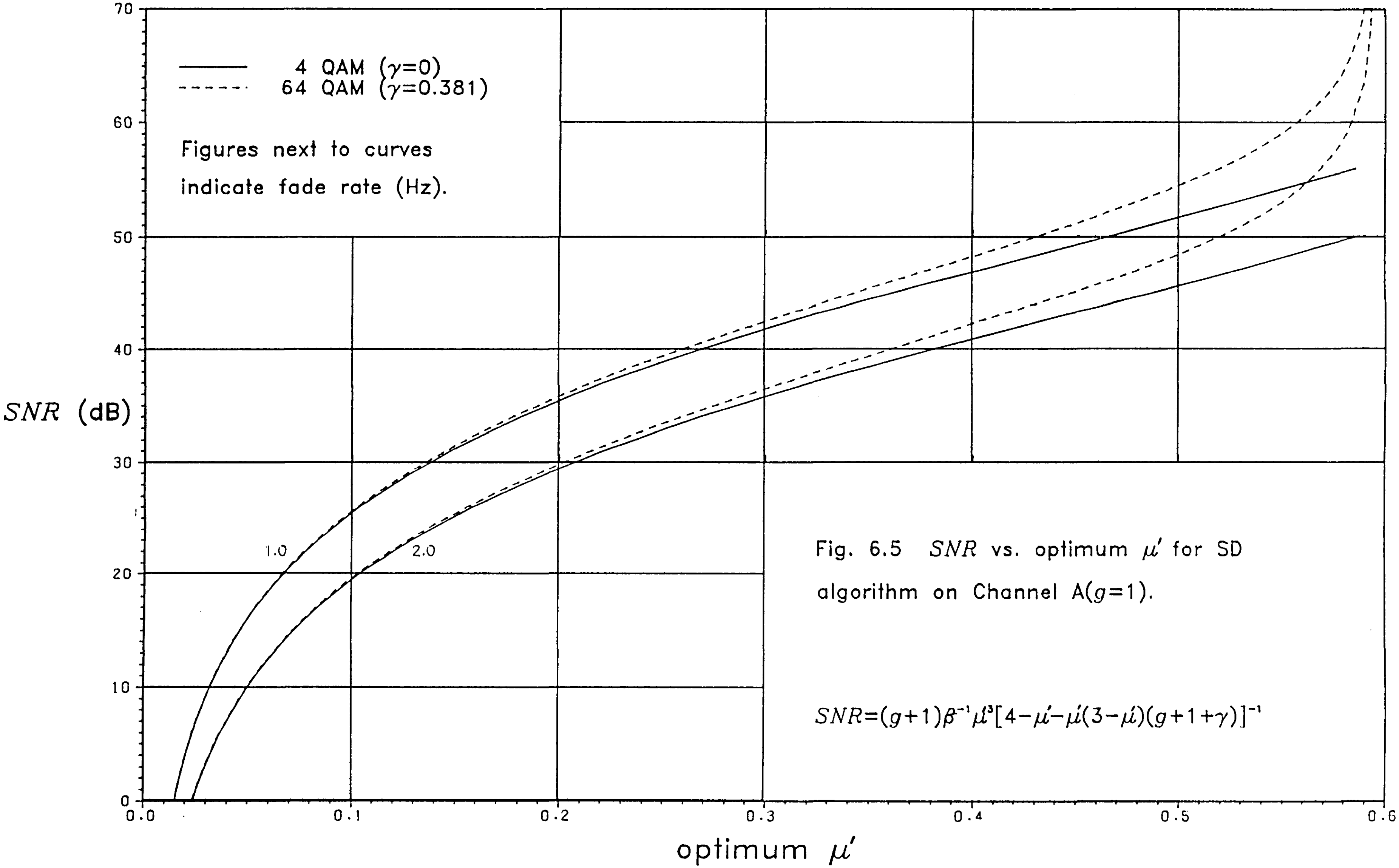
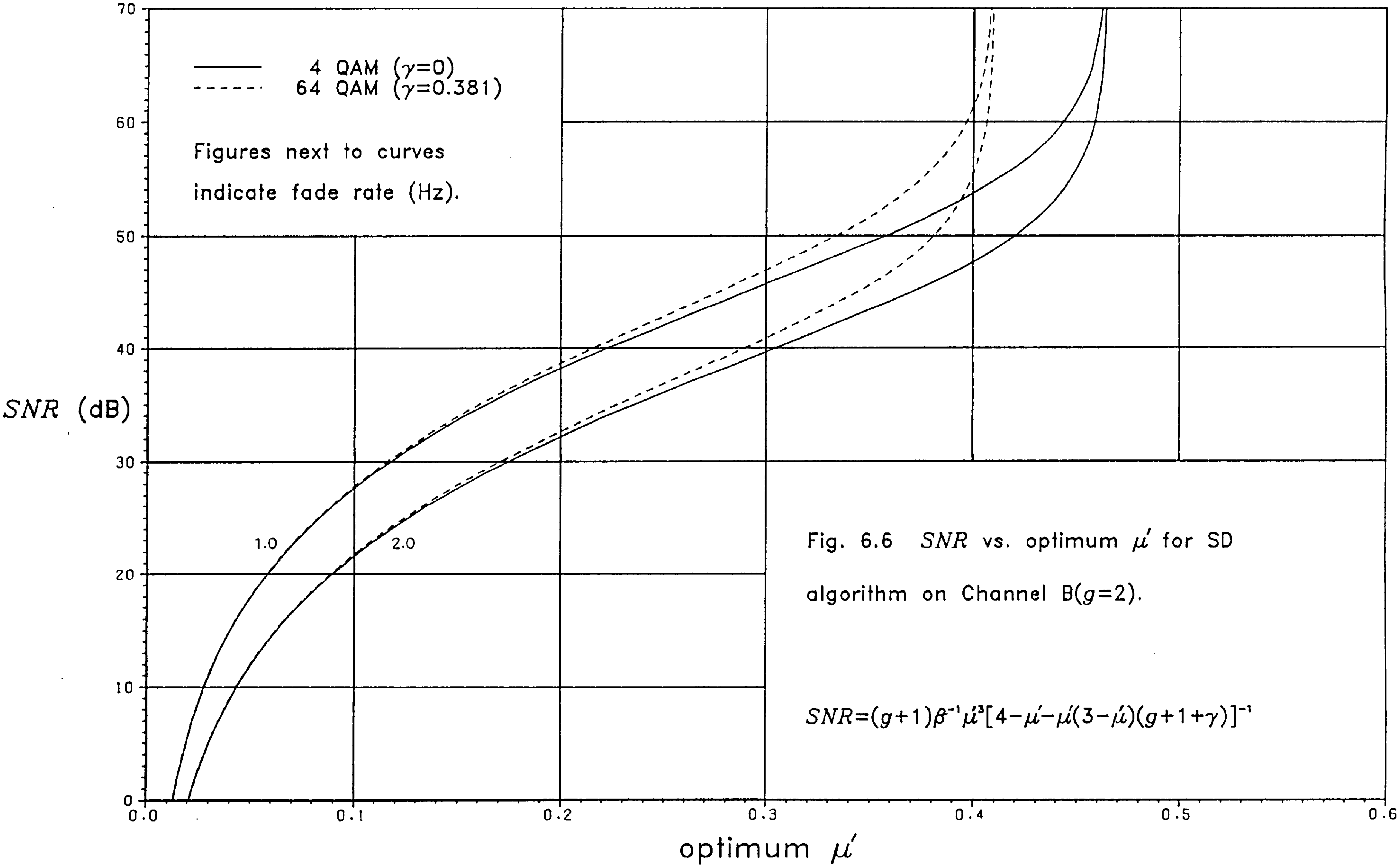


Fig. 6.5 SNR vs. optimum μ' for SD algorithm on Channel A(g=1).

$$SNR=(g+1)\beta^{-1}\mu'^{3}[4-\mu'-\mu'(3-\mu')(g+1+\gamma)]^{-1}$$



Figures next to curves indicate fade rate (Hz).

Fig. 6.6 *SNR* vs. optimum μ' for SD algorithm on Channel B(g=2).

$$SNR=(g+1)\beta^{-1}\mu'^{3}[4-\mu'-\mu'(3-\mu')(g+1+\gamma)]^{-1}$$

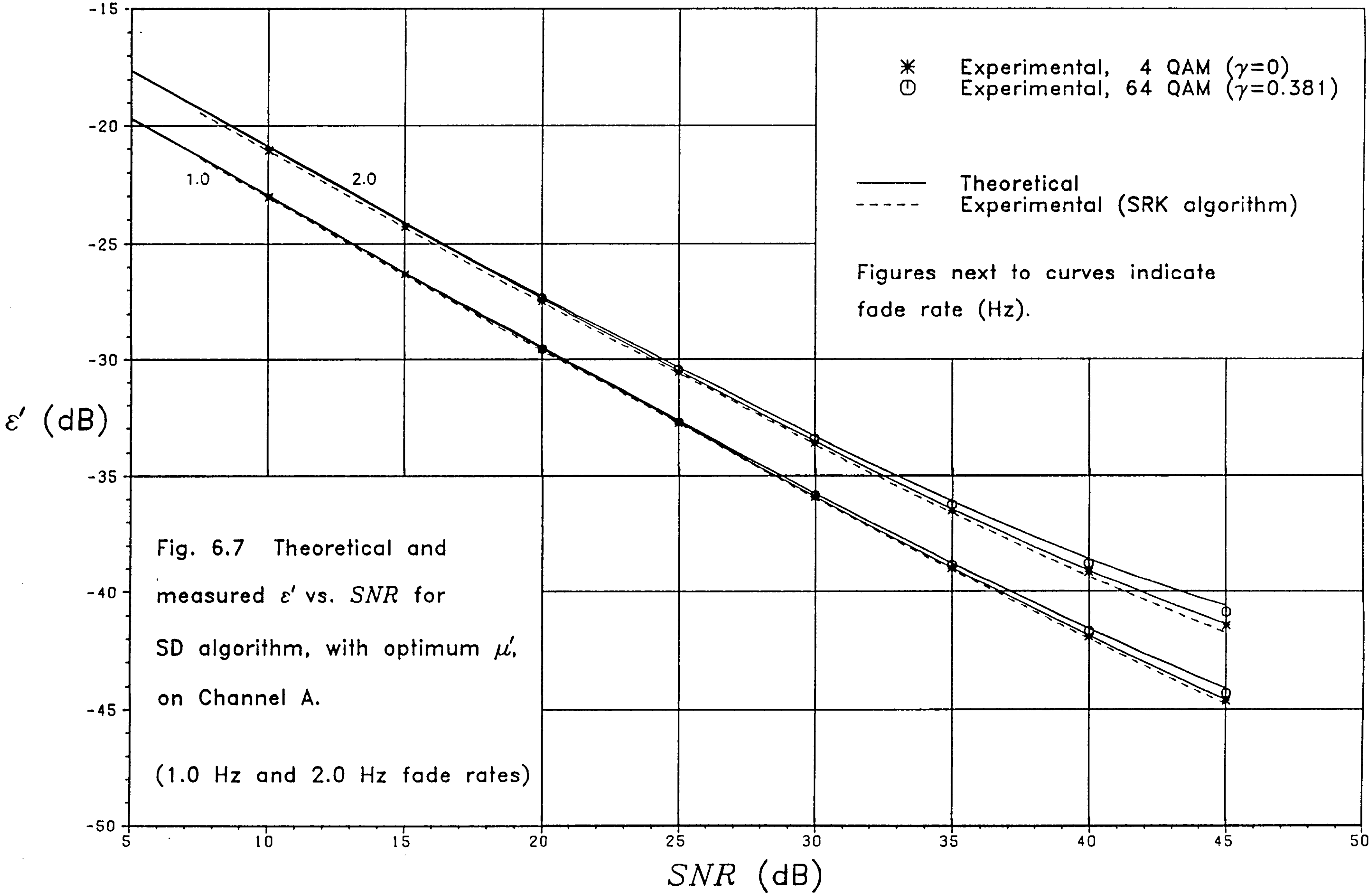


Fig. 6.7 Theoretical and measured ε' vs. SNR for SD algorithm, with optimum μ', on Channel A.

(1.0 Hz and 2.0 Hz fade rates)

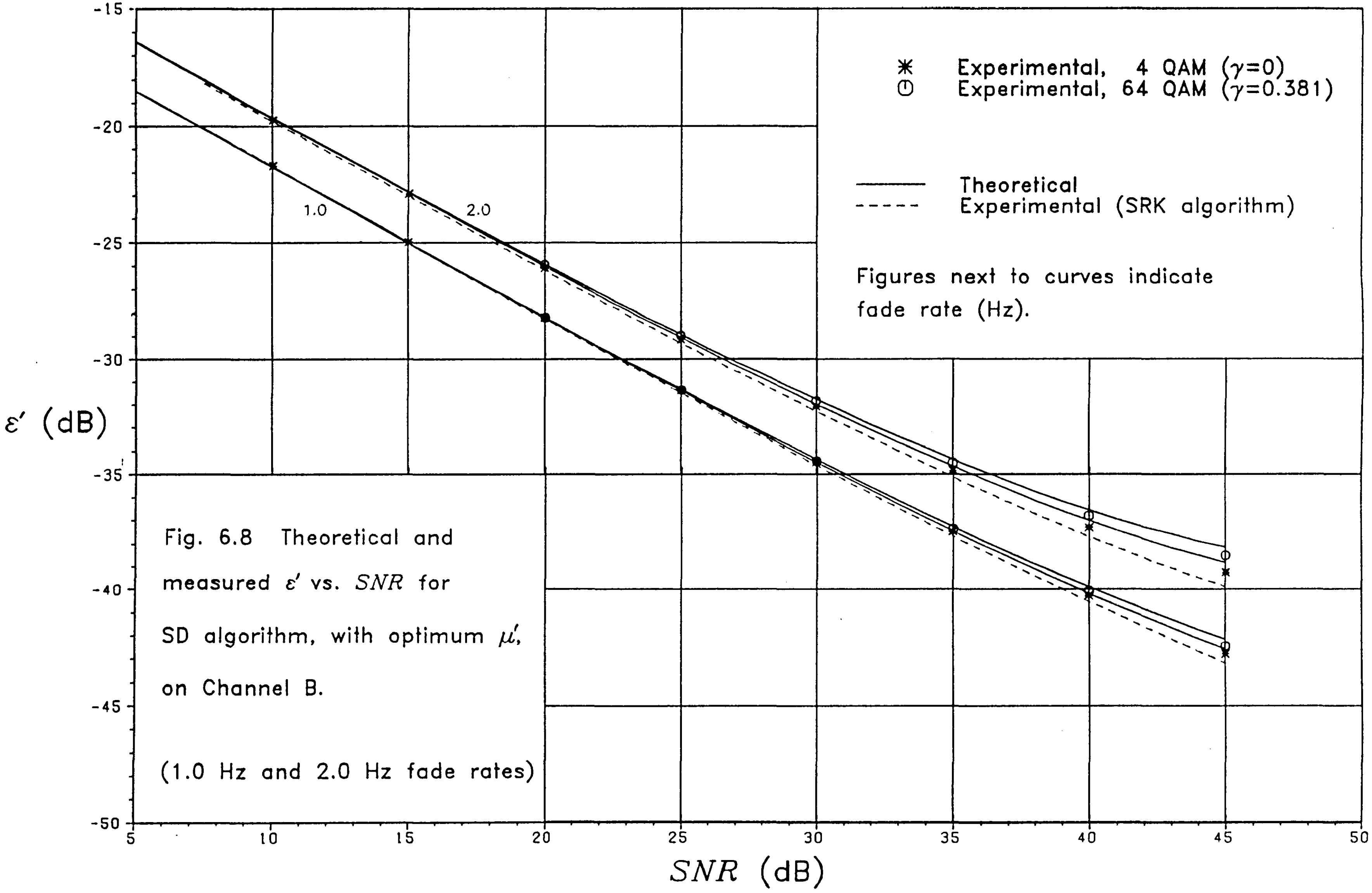


Fig. 6.8 Theoretical and measured ε' vs. SNR for SD algorithm, with optimum μ', on Channel B.

(1.0 Hz and 2.0 Hz fade rates)

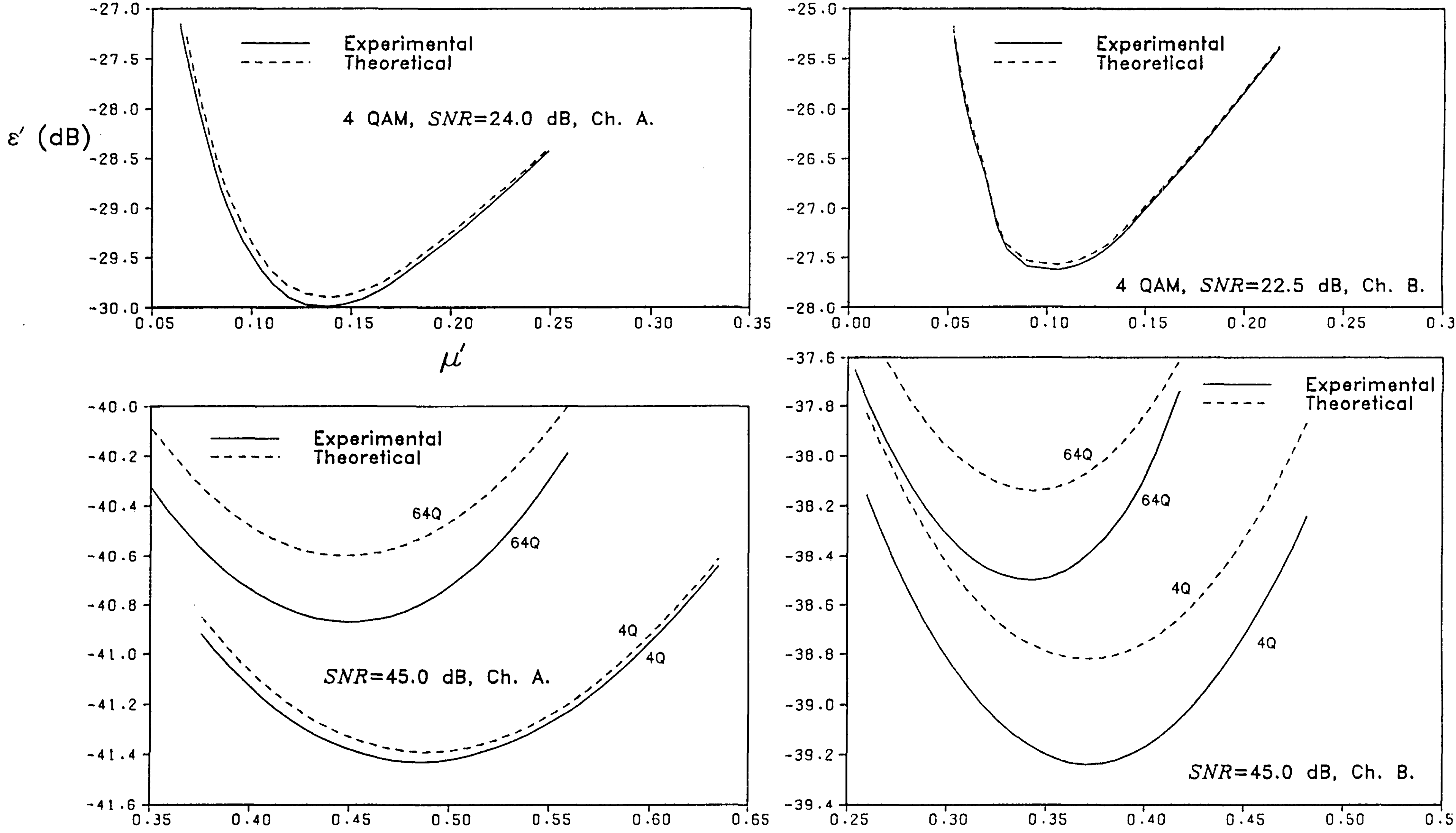


Fig. 6.9 Variation of ε' with μ' for SD algorithm, f.r.=2.0 Hz.

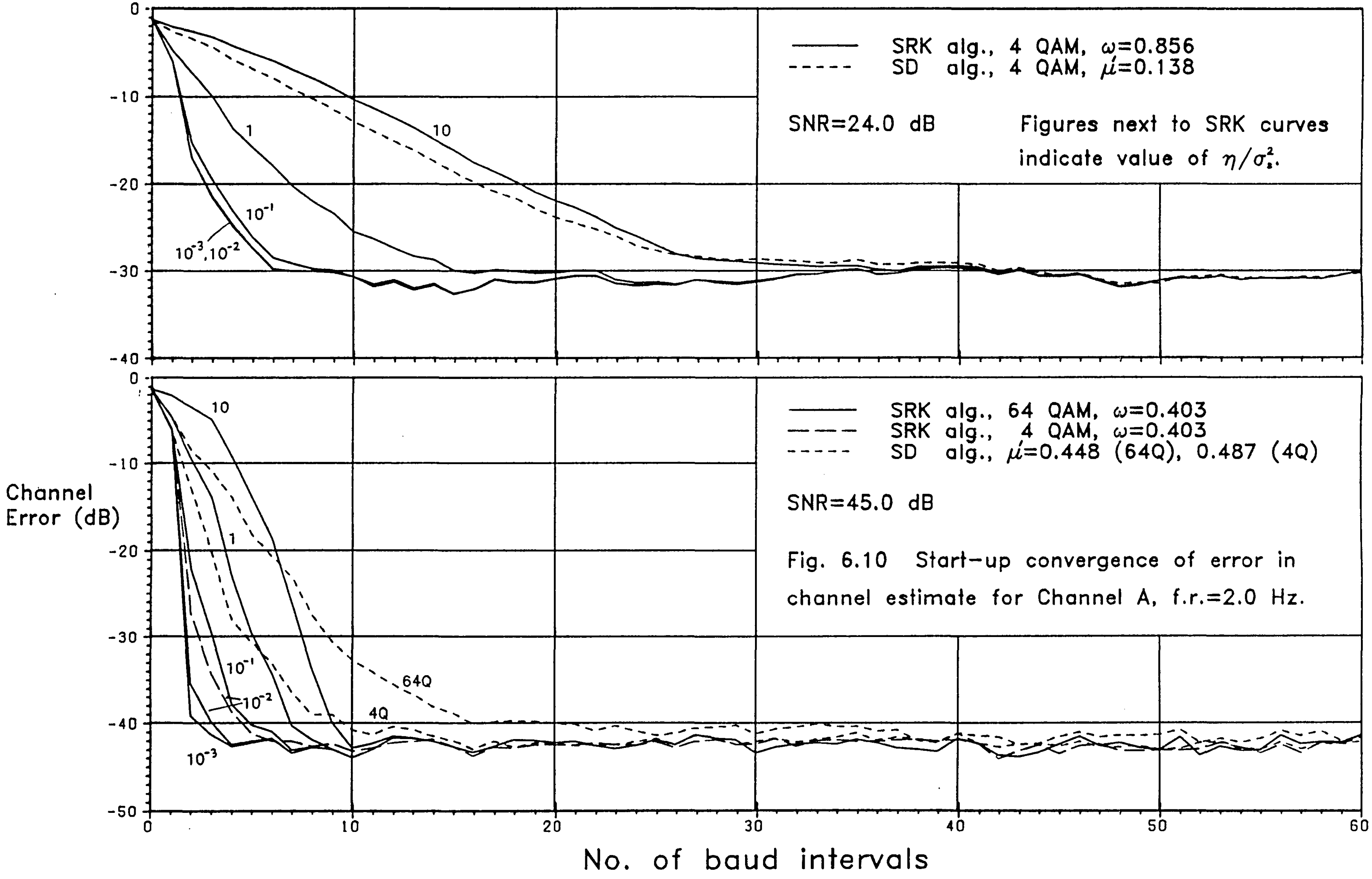


Fig. 6.10 Start-up convergence of error in channel estimate for Channel A, f.r.=2.0 Hz.

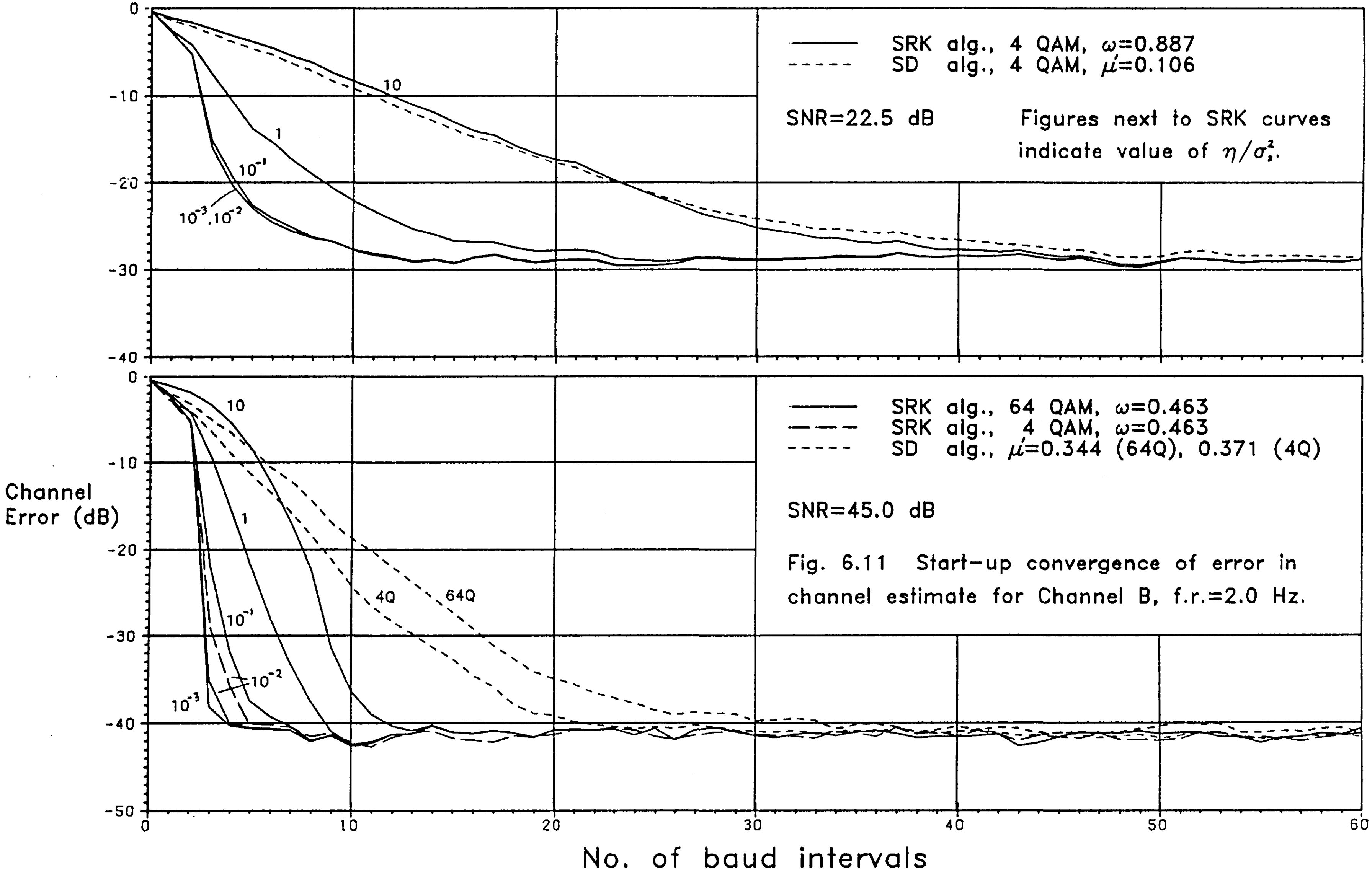


Fig. 6.11 Start-up convergence of error in channel estimate for Channel B, f.r.=2.0 Hz.

Chapter 7
MODE II PERFORMANCE

In this chapter the performance of receiver devices is observed when the channel variations have to be followed by a tracking algorithm, subject to the condition that receiver decisions it uses are always correct.

We will first describe how the DFE taps can be computed from the channel estimate at a cost less than the conventional SRK approach to the DFE, and then show that this alternative tracking procedure gives a superior performance as well. A modified form of the SRK algorithm by Hsu [16] is also shown to be prone to instability under certain conditions. It is demonstrated that the SD algorithm competes favourably with the SRK when used to implement the DFE by way of the channel estimate, and from the results in chapter 6 it is shown how simple theoretical predictions can be made of the performance of the DFE-II$_1$. Finally, P_e(E.C.) results are presented for the DFE and MLSE/near-MLSE and merits discussed.

7.1 Computation of DFE Taps from Channel Estimate

Recall from section 2.6.1 that the DFE taps at time kT are given as

$$\underline{\mathbf{w}}_k = \mathbf{Y}_k^{*-1}\underline{\mathbf{y}}_{k0}^* \tag{7.1}$$

$$b_{ki} = \underline{\mathbf{w}}_k^{\mathrm{t}}\underline{\mathbf{y}}_{k,-i} \qquad \text{for } 1 \leq i \leq g \tag{7.2}$$

with $\mathbf{Y}_k$ given as

$$\mathbf{Y}_k = \sum_{i=0}^{N-1}\underline{\mathbf{y}}_{ki}\underline{\mathbf{y}}_{ki}^{*\mathrm{t}} + \rho\mathbf{I} \tag{7.3}$$

and $\underline{\mathbf{y}}_{ki}$ as defined in (2.129). Let $\hat{\rho}$ be a positive real quantity, about which we will discuss later, but for the moment can be assumed, like ρ, to be small. Let us now form an approximation to $\mathbf{Y}_k$ from the elements of the channel estimate vector $\underline{\mathbf{c}}_{k-1}$, defining it as

$$\hat{\mathbf{Y}}_k = \sum_{i=0}^{N-1}\underline{\mathbf{c}}_{k-1,i}\underline{\mathbf{c}}_{k-1,i}^{*\mathrm{t}} + \hat{\rho}\mathbf{I} \tag{7.4}$$

where $$\underline{c}^{t}_{k-1,i} = [c_{k-1,1-i}\ c_{k-1,2-i} \cdots c_{k-1,N-i}] \tag{7.5}$$

the elements $c_{k-1,i}=0$ for $i<1$ and $i>g+1$. Using $\hat{\mathbf{Y}}_k$, an estimate of the MMSE DFE taps is given as

$$\hat{\underline{\mathbf{w}}}_k = \hat{\mathbf{Y}}_k^{*-1}\underline{c}^{*}_{k-1,0} \tag{7.6}$$

$$\hat{b}_{ki} = \hat{\underline{\mathbf{w}}}^{t}_{k}\underline{c}_{k-1,-i} \qquad \text{for } 1\leq i\leq g \tag{7.7}$$

At first sight it might appear that to calculate $\hat{\underline{\mathbf{w}}}_k$ we have to invert the matrix $\hat{\mathbf{Y}}^{*}_{k}$. Using the fact that $\hat{\mathbf{Y}}_k$ is a positive definite Hermetian matrix, this can be accomplished with multiplications proportional to N^3 [86]. However, by exploiting certain properties of $\hat{\mathbf{Y}}_k$ we show it is possible to calculate $\hat{\underline{\mathbf{w}}}_k$ with multiplications proportional to N^2. Observe that $\hat{\mathbf{Y}}^{*}_{k}$ can be expressed as

$$\hat{\mathbf{Y}}^{*}_{k} = \mathbf{L}^{*}_{k}\mathbf{L}^{t}_{k}+\hat{\rho}\mathbf{I} \tag{7.8}$$

where $\mathbf{L}_k$ is a lower triangular matrix given by

$$\mathbf{L}_k = [\underline{c}_{k-1,0}\ \underline{c}_{k-1,1} \cdots \underline{c}_{k-1,N-1}] \tag{7.9}$$

Assuming $c_{k-1,1}\neq 0$, $\mathbf{L}_k$ possesses a unique inverse $\mathbf{L}_k^{-1}$. It is easy to show that

$$\mathbf{L}_k^{*-1}\underline{c}^{*}_{k-1,0} = [1\ 0 \ldots 0]^{t} \tag{7.10}$$

Then to get $\hat{\underline{\mathbf{w}}}_k$ we need to solve the equation

$$(\mathbf{L}^{t}_{k}+\hat{\rho}\mathbf{L}_k^{*-1})\hat{\underline{\mathbf{w}}}_k = [1\ 0 \ldots 0]^{t} \tag{7.11}$$

The matrix $\mathbf{L}_k$ is Toeplitz, i.e. the elements along any northwest-southeast diagonal are equal. It is shown in Appendix H that $\mathbf{L}_k^{*-1}$ is also Toeplitz. Thus the matrix $(\mathbf{L}^{t}_{k}+\hat{\rho}\mathbf{L}_k^{*-1})$ is Toeplitz. This fact enables us to solve (7.11) by applying the method of the Levinson-Durbin algorithm [3], [87] for solving Yule-Walker systems. Appendix H gives full details of the procedure and of an algorithm to solve for the DFE taps.

Note that the parameter $\hat{p}$ cannot be set to zero, because then we would be solving

$$\mathbf{L}_k^t \underline{\hat{\mathbf{w}}}_k = [1 \; 0 \ldots 0]^t \tag{7.12}$$

which yields, with N finite and $c_{k-1,1} \neq 0$, the trivial solution

$$\underline{\hat{\mathbf{w}}}_k = [\frac{1}{c_{k-1,1}} \; 0 \ldots 0]^t \tag{7.13}$$

With N infinite there is more than one solution to (7.12), the ZF DFE(∞,g) case being one of them, because it is now possible to have $\hat{w}_{k,N-2} > \hat{w}_{k,N-1} \rightarrow 0$ as $N \rightarrow \infty$.

Table 7.1 gives a comparison of the computational and storage requirements that arise when the implementation of the adaptive DFE is by way of the channel estimate (in future to be referred to as the channel estimate (CE) method) and by way of the conventional method with the SRK algorithm. Note that in addition to the computations involved in solving for the DFE taps (as given in Appendix H) we also, for the CE method, have to include the computations needed to produce the channel estimate, which is assumed to be via the SRK algorithm. The storage requirement for the CE method is the sum of that for producing the channel estimate from the SRK algorithm and that for the computation of the DFE taps, less $2(g+1)$ because the elements $\{l_i\}$, $1 \leq i \leq (g+1)$, in Appendix H are also the components of the channel estimate, which are the elements $\{c_i\}$ in Appendix C. Actually the storage requirement for the CE method can be even less, because some of the "working" variables used in the SRK algorithm can be used again in the computation of the DFE taps. The exact requirements may vary slightly from those in Table 7.1 for different signal processors.

Table 7.1 shows that, except for 2 extra real reciprocals, the CE method is more economic than the conventional method, and the numerical advantage increases with increasing N, g. The CE method does, however, require more organisation than the conventional method, but this should not be a restrictive factor.

Requirement	CE Method	Conventional Method
Equivalent Real Multiplications per Iteration.	$6(N+g)^2+11(N+g)$ $-2g(2N+g-3)$ $-N+11$ e.g. DFE(6,1): 356 DFE(6,2): 433	$6(N+g)^2+11(N+g)$ e.g. DFE(6,1): 371 DFE(6,2): 472
Equivalent Real Additions per Iteration.	$6(N+g)^2+4(N+g)$ $-2g(2N+g-3)$ $-8N+12$ e.g. DFE(6,1): 266 DFE(6,2): 336	$6(N+g)^2+4(N+g)$ e.g. DFE(6,1): 322 DFE(6,2): 416
Equivalent Real Reciprocals per Iteration.	$N+g+2$ e.g. DFE(6,1): 9 DFE(6,2): 10	$N+g$ e.g. DFE(6,1): 7 DFE(6,2): 8
Equivalent Real Variable Storage Locations.	$< (N+g)^2+9(N+g)$ $+14-N(N+2g+3)$ $+4g+17$ e.g. DFE(6,1): <81 DFE(6,2): <97	$(N+g)^2+9(N+g)$ $+14$ e.g. DFE(6,1): 126 DFE(6,2): 150

Table 7.1 Comparison of requirements for implementing adaptive DFE by way of CE method and conventional method.

7.2 Performance Comparison of CE and Conventional Methods

Start-up convergence will be discussed first. This is an important consideration when periodic training is employed, as convergence is required in the shortest number of baud intervals.

The parameter $\hat{\rho}$ in the CE method should be small, and ideally we expect it should equal the quantity ρ. In the lower halves of figs. 7.1–7.4 are plotted the start-up convergences of the MSE in detection, ξ_k, for the DFE(6,g), produced by the CE method for SNR=22.5 dB and 40.0 dB with 4 QAM and 64 QAM respectively, and with $\hat{\rho}$ set at various values. The parameter η is set to the value $10^{-2}\sigma_s^2$, which we saw in chapter 6 gives a convergence rate close to optimum. The fade rate in all cases is 1.0 Hz, and in the time period of 60 baud intervals the channel response vector has changed from $(-0.638-j0.220\ ,\ -0.409+j0.327)$ to $(-0.648-j0.229\ ,\ -0.373+j0.404)$ for Channel A, and from $(-0.521-j0.180\ ,\ -0.334+j0.267\ ,\ -0.636-j0.059)$ to $(-0.529-j0.187\ ,\ -0.305+j0.330\ ,\ -0.633-j0.074)$ for Channel B. Initially the DFE taps (both feedforward and feedback) are zero, as is also the initial channel estimate vector $\underline{c}_{-1}$. The starting conditions in the SRK algorithm are as given in Appendix C, and the input vector $\underline{x}_0$ consists of data symbol values. Each convergence curve is the average of 30 separate runs, each run involving a different sequence of noise samples and data symbols. The parameter ω is set to the theoretically optimum value from fig. 6.1 that minimizes the long-term error in the channel estimate.

The convergence curves for the different values of $\hat{\rho}$ all appear to level out at the same time, and the ones for $\hat{\rho}=\rho$ always reach the lowest steady-state level. From theory we know that the smaller ρ is, the closer the MMSE DFE(∞,g) is to the ZF DFE(∞,g). Therefore, as long as $\hat{\rho}$ is just a small enough number, there should be little change in performance for variation in $\hat{\rho}$, as the convergence curves bear out. The curves support the expectation that $\hat{\rho}=\rho$ is ideal, and also suggest that if $\hat{\rho}\neq\rho$ then it is better to have $\hat{\rho}<\rho$ than $\hat{\rho}>\rho$. In the rest of this thesis employment of the CE method is made with the setting $\hat{\rho}=\rho$.

The upper halves of figs. 7.1–7.4 depict the convergence curves for the conventional method under the same conditions as the lower halves for the CE method, with η/σ_s^2 varying between 10^{-4} and 10.0. The DFE taps, which form the elements of the estimated vector, are initially zero. The starting conditions in the SRK algorithm are as given in Appendix C, and the input vector $\underline{\mathbf{x}}_0$ consists of N received samples and g data symbols (recall (3.46) of chapter 3). The curves are again formed by averaging over 30 different sequences of noise samples and data symbols. The curves indicate that $\eta/\sigma_s^2{\sim}10^{-3}$–$10^{-2}$ gives a convergence rate close to optimum. The optimum value of η must depend to some extent on the channel response vector, because of the latter's influence on the input vector; note that this is not the case for channel estimation. The values of ω used in the SRK algorithm have been optimized for long-term performance, about which more will be said later.

Table 7.2 gives a comparison of the convergence times of the CE and conventional methods, the information being obtained form figs. 7.1–7.4. The convergence time is taken as the number of whole baud intervals required by a curve to reach within 2.0 dB of its approximate steady-state value. The values given in the table are for the $\hat{\rho}=\rho$ curves in the CE method and the "fastest" curves in the conventional method.

Channel conditions (constellation size, channel, *SNR* (dB))	Time, in baud intervals, to reach specified dB value of ξ_k from start-up.	
	CE method	Conventional method
4, A, 22.5	5 for −14 dB	11 for −13 dB
64, A, 40.0	4 for −17 dB	9 for −14 dB
4, B, 22.5	5 for −13 dB	12 for −12.7 dB
64, B, 40.0	4 for −14 dB	11 for −13 dB

<u>Table 7.2</u> Comparison of convergence times for CE and conventional methods.

Table 7.2 indicates that the CE method is about 2–3 times faster than the conventional method. The reason for this will be discussed later, after we have looked at the long-term performances over the 25 fade period patterns.

In the CE method, the parameter ω of the SRK algorithm is set to that value which theoretically minimizes the error in the channel estimate. With the conventional method, the calculation of the optimum ω setting that minimizes, say, the average mean square error ξ is not easy. We resort, therefore, to determining it by measurement. Fig. 7.5 gives the optimum ω, for the conventional method, that effectively minimizes ξ and P_e in mode II_1 at a fade rate of 1.0 Hz. Each plot point involved minimizing ξ with respect to ω, a "minimization curve" being formed by measuring ξ at different settings of ω at a particular *SNR*, each measurement involving a run of 60,000 data symbols over the fading patterns, with a warm-up time of 1.67 fade periods to eliminate start-up transients. It was found that the optimum ω at a particular *SNR* is virtually independent of the constellation size, so fig. 7.5 is valid for 4, 16 and 64 QAM. It was also noticed that the optimum ω virtually minimizes the analytical P_e estimate too, this being almost exact for error rates above 10^{-4}, and for the lower error rates observed the difference between the minimum P_e and that with ξ-optimized ω is negligible.

Figs. 7.6–7.9 compare the performances of the two adaptive methods in mode II_1, both methods having a warm-up time of 1.67 fade periods to eliminate start-up transients. The graphs show that the CE method yields a useful improvement over the conventional method, this being larger for increasing constellation size. For example, on Channel A at $P_e=10^{-3}$, the advantage in *SNR* is about 0.9, 1.4 and 3.4 dB respectively for 4, 16 and 64 QAM, whilst at $P_e=10^{-5}$ it is about 1.5 and 3.2 dB for 4 and 16 QAM respectively. Also, on Channel B at $P_e=10^{-3}$, the advantage is about 0.5, 1.0 and 1.6 dB respectively for 4, 16 and 64 QAM, whilst at $P_e=10^{-5}$ it is about 0.9 and 1.6 dB for 4 and 16 QAM respectively. The advantage of the CE method appears quite significant for 64 QAM, being greater than 3.0 dB beyond $P_e=0.5\times10^{-4}$. Both the P_e and ξ criteria indicate that the advantage of the CE method increases with increasing *SNR*. In fact, with 16 QAM on Channel A, and with 64 QAM on both Channel A and Channel B, the P_e and ξ curves for the conventional method show a distinct tendency to start levelling out to an irreducible level, indicating that the effect of ISI is becoming quite influential.

Observe in Table 7.3 the *SNR* advantage of the CE method, i.e. the additional amount of *SNR* needed to enable the conventional method to perform at the same level as the CE method at a given *SNR*, for both the P_e and ξ performance criteria, the information being taken from figs. 7.6–7.9.

SNR	Channel A						Channel B					
	4 QAM		16 QAM		64 QAM		4 QAM		16 QAM		64 QAM	
(dB)	P_e	ξ	P_e	ξ	P_e	ξ	P_e	ξ	P_e	ξ	P_e	ξ
10.0	0.6	0.5					0.4	0.4				
15.0	0.7	0.6					0.4	0.5				
20.0	0.8	0.7	0.6	0.7			0.5	0.7	0.5	0.7		
25.0	1.1	1.1	0.9	1.0			0.9	1.0	0.7	1.0		
30.0	1.4	1.6	1.2	1.5	1.4	1.5			1.1	1.5	1.0	1.5
35.0			1.8	2.5	2.3	2.5			1.9	2.4	1.5	2.4
40.0			3.3	4.3	3.6	4.5					3.0	4.3

Table 7.3 *SNR* advantage (dB) of CE method over conventional method, using P_e and ξ criteria.

From the values in Table 7.3, the ξ criterion, on the whole, appears to indicate a larger advantage for the CE method than does the P_e criterion. The difference between the criteria is slight for low *SNR*, getting larger, but not excessively so, as the *SNR* increases. The fact that there is sometimes a difference between the two criteria, errors in measurement not withstanding, implies that we can have a situation where the two adaptation methods, operating under appropriate *SNR*'s, can yield the same P_e but different ξ's and vice-versa. This is not altogether unexpected, because P_e is a quantity that is also influenced by the variance (over time) of the instantaneous mean square error ξ_k, as well as by its time-average ξ. For the reason that P_e is a more informative indicator of performance, the advantage expressed by this criterion is to be preferred. A point to remember, however, in favour of the ξ

criterion is from the results of chapter 5, where it was demonstrated (for a ZF MFE in mode I_1) that ξ, measured over the limited fading patterns, is more representative of its corresponding theoretical long-term value than is P_e.

It was noted in the last section that we need $c_{k-1,1} \neq 0$ for $\mathbf{L}_k$ to possess a unique inverse $\mathbf{L}_k^{-1}$; the reciprocal of $c_{k-1,1}$ is used in the CE method. In practice, therefore, $c_{k-1,1}$ should never be zero, and there should be a lower bound set for $|c_{k-1,1}|$ that is the smallest "safe" number allowable with the numerical precision available. For the simulations conducted in this thesis, however, the only constraint applied was that if $c_{k-1,1}=0$ the DFE taps are not updated; this is not to say that the constraint ever came into play. Several simulations were run with (7.6) solved by a more standard routine, such as the one in [86] (based on Cholesky decomposition), and compared with the simulations from the CE method. The computer uses floating point arithmetic with 14 decimal digits for the mantissa; using a measurement accuracy of 8 significant digits, the agreement between the method using the process of [86], which is more costly in terms of computations, and the CE method, was exact. If more limited precision processors are employed, another strategy to use if $|c_{k-1,1}|$ becomes too small is to simply set the first tap of the feedforward filter to zero, and compute the rest of the taps assuming $c_{k-1,2}$ as the first component in the channel vector.

The method of implementing the MMSE DFE via the channel estimate could perhaps be made even more computationally efficient, especially when N, g are large. This is because the channel estimate is not expected to change too much in successive baud intervals. Therefore, some cost-reducing iteration scheme could perhaps be devised to compute the new MMSE tap settings from the new channel estimate and old tap settings and channel estimate.

We have demonstrated that the CE method is superior to the conventional SRK method in both computational economy and performance. A pertinent question to ask is why does it perform better. The faster convergence can be easily explained. In the conventional method the length of the estimated vector in the RLS algorithm is $N+g$, whereas for channel estimation the length is $g+1$. It was discussed in chapter 3 how the speed of convergence of the RLS algorithm is dependent on the length of the estimated vector, the longer the vector the slower the convergence. Thus since the speed of convergence of the CE method is determined solely by how fast the channel estimate can be formed, the CE method will be faster than the conventional method. As to why the CE method has a better long-term performance, recall first that the estimated vector $\underline{c}_k$ in the RLS algorithm is given by

$$\underline{c}_k = (\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^*\underline{x}_i^{t})^{-1}(\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^* z_i) \tag{7.14}$$

where for the conventional method $\underline{x}_i$ and z_i are

$$\underline{x}_i^{t} = [r_i \; r_{i+1} \cdots r_{i+N-1} \; \hat{s}_{i-1} \; \hat{s}_{i-2} \cdots \hat{s}_{i-g}] \tag{7.15}$$

$$z_i = \hat{s}_i \tag{7.16}$$

while for the CE method, which uses channel estimation, they are

$$\underline{x}_i^{t} = [\hat{s}_i \; \hat{s}_{i-1} \cdots \hat{s}_{i-g}] \tag{7.17}$$

$$z_i = r_i \tag{7.18}$$

Rearranging (7.14) we obtain

$$\sum_{i=0}^{k}\omega^{k-i}\underline{x}_i^*(\underline{x}_i^{t}\underline{c}_k - z_i) = \underline{0} \tag{7.19}$$

Assuming there are no decision errors in the $\{\hat{s}_i\}$, it can be seen from (7.19) that in order to have a unique value for $\underline{c}_k$, the value of k needs to be $\geq M-1$, where M is the length of the estimated vector, with at least M linearly independent $\underline{x}_i$'s which, for good adaptation, should be the most recent. Consequently, the conventional method needs at least $2N+g-1$ noise-contaminated received samples r_i (these being in the $N+g$ successive input vectors $\underline{x}_i$), whereas the CE method needs only $g+1$ (these being in the $g+1$ desired values z_i), and in a time-varying environment the accuracy of the conventional method will further suffer because of it's need for a larger span of received samples. Thus it is not surprising that the conventional method is inferior to the CE method. Furthermore, if the $\mathbf{P}_k^{-1}$ matrix in the RLS algorithm is ill-conditioned (more likely if ω is smaller), then small errors in the elements of $\mathbf{P}_k^{-1}$, as can be produced by noise in the $\{\underline{\mathbf{x}}_i\}$, lead to larger errors in the inverse $\mathbf{P}_k$.

The CE method is a more determined attempt at implementing the MMSE DFE, taking advantage of the special relationship between the optimum DFE tap settings and the channel response, and the accuracy of the channel estimate is thus the crucial issue; the conventional method simply determines the DFE tap settings that minimize a weighted time average of the squared error at the input to the threshold detector.

To conclude this section we mention a recent development in [88]. In this paper it is described how roots of the z-transform of the sampled channel impulse response, which are outside the unit circle, can be calculated in an iterative manner. These roots are then used to evaluate the leading N taps of the feedforward section of the ZF DFE, which is the pre-filter for the near-MLSE, the feedforward filter having a z-transform which has poles that are the roots in question, and zeros that are the reciprocals of the complex conjugates of these roots (recall end of section 2.5). Because N is finite-valued, there will be some anti-causal ISI present at the output of the feedforward filter; therefore N has to be made sufficiently large to limit this. Roots that are close to the unit circle require a large value of N in order for them to be accurately accommodated by the filter [88]. At the same time, however, the closer to the unit circle the roots are, the less urgent the need for them to be accommodated. It is suggested in [88] that the scheme be adopted for use on time-varying channels, by employing an estimate of the channel response. The fundamental difference between an approach based on the scheme in [88] and the CE method, is that the former aims to produce a truncated-tap version of the ZF DFE(∞,g) from the channel estimate, whereas the latter produces a finite-tap MMSE DFE(N,g), which is optimum for the given value of N. With N large enough, and $\hat{p}$ small, and ignoring any differences concerning the degree of numerical precision required, in theory one would expect there to be little difference between the levels of performance offered by the two approaches, the possible advantage of one approach over the other being in terms of economy of computations and ease of implementation. In general, the computational requirements of the scheme in [88] depend significantly on the number of roots outside the unit circle, and the number of iterations needed to locate them to within a given accuracy.

7.3 Revised SRK Algorithm of Hsu [16]

It has been observed how the parameter η in the SRK algorithm controls convergence; a large value produces a slow rate. In [16] it is pointed out that a small value for this parameter (identified as "ξ" in [16]) produces a large magnitude disparity in the start-up computations of the diagonal matrix $\mathbf{D}_k$. To be more specific, it states for example how the first component of this matrix drops from 1.0 to about 10^{-3} in the first iteration, the other components staying at about 1.0, this example being for the conventional implementation of a DFE with $\eta \sim 10^{-3}$ and ω close to 1. For a processor of limited word size, this large magnitude disparity can significantly disturb the equalization process during start-up, and also if a periodic resetting scheme is used (the resetting is for preventing the cumulation of round-off errors). To overcome this situation a modified version of the SRK algorithm is presented in [16], and is such that it is "stable" for various values of η without sacrificing the rate of convergence. The essential difference lies in the replacement of the step

$$\mathrm{P}_k = \tfrac{1}{\omega}(\mathrm{I}-\underline{\mathrm{k}}_k\underline{\mathrm{x}}_k^{\mathrm{t}})\mathrm{P}_{k-1} \tag{7.20}$$

as was in (3.42iii) of chapter 3, with

$$\mathrm{P}_k = (\tfrac{1}{\omega}\mathrm{I}-\underline{\mathrm{k}}_k\underline{\mathrm{x}}_k^{\mathrm{t}})\mathrm{P}_{k-1} \tag{7.21}$$

What this means to the algorithm in Appendix C is that we replace those computational stages marked on the left-hand side with "Δ", namely

$$\left.\begin{array}{ll}
(i) & \kappa = \frac{1}{\alpha_1} \\
(ii) & d_1 = d_1 h_\omega \eta \kappa \\
(iii) & \beta_0 = \alpha_{i-1} \\
(iv) & \kappa = \frac{1}{\alpha_i} \\
(v) & e = \epsilon_k \kappa
\end{array}\right] \tag{7.22}$$

with, in respective order,

$$\left.\begin{array}{ll} (i) & \kappa = \dfrac{1}{(\alpha_1 + h_t)} \\ (ii) & d_1 = d_1 h_\omega (\eta + h_t)\kappa \\ (iii) & \beta_0 = \alpha_{i-1} + h_t \\ (iv) & \kappa = \dfrac{1}{(\alpha_i + h_t)} \\ (v) & e = \epsilon_k \kappa h_\omega \end{array}\right] \tag{7.23}$$

where
$$h_t = \alpha_M (h_\omega - 1) \tag{7.24}$$

The mathematical proof of the steps in (7.23) can be found in [16]. For ω close to 1, there is little difference between the revised and original SRK algorithms. In [16] the revised SRK algorithm is observed to be superior, though only very marginally so, to the original SRK, and this is put down to the improvement of stability of the revised SRK. We have found, however, that for ω below a certain value the revised SRK algorithm becomes unstable, and that the threshold for this is dependent on the length of the estimated vector. Observe in Table 7.4 some error count performance figures for the conventional implementation of the adaptive DFE by means of the revised SRK algorithm. The figures were obtained over a time period that covers only part of the 25 fade period pattern of Channel B, after an initial warm-up time of 2000 baud intervals to remove transient effects.

It was found that the sudden large jump in the error count, that occurs on reaching a particular setting of $1/\omega$, is caused by the real positive elements of $\mathbf{D}_k$ increasing at each iteration, until they approach the upper limit for a positive number on the computer, which is $\sim 10^{322}$. In general, if any number exceeds the upper limit the program aborts. This did not happen here, because before the elements $\{d_i\}$, or the elements $\{\mathrm{g}_i\}$ and $\{\alpha_i\}$ as well (see Appendix C), could reach the upper limit, the quantity κ reached zero, it being the reciprocal of a very large number. This then caused the $\{d_i\}$ to become zero, and thence on the next update the elements $\{\mathrm{g}_i\}$ all became zero and $\{\alpha_i\}$ all became equal to η. Thus tracking ceased altogether, the estimated vector becoming "frozen" at a constant value.

$\frac{1}{\omega}$	No. of errors in 20,000 data symbols (64 QAM)				
	DFE(15,2), *SNR*=100.0 dB		DFE(10,2), *SNR*=100.0 dB		DFE(6,2)
	f.r.=0.5 Hz (over 40,000)	f.r.=1.0 Hz	f.r.=1.5 Hz	f.r.=2.0 Hz	f.r.=1.0 Hz *SNR*=40.0 dB
1.050	9	204			
1.075	0	84			47
1.100	0	43	72		25
1.125	1	28	43	134	15
1.150	7,862	26	35	101	12
1.175	23,363	6,703	27	79	10
1.200		10,436	1,622	1,232	10
1.225			10,535	9,417	12
1.250				13,600	12
1.275					7,437

N	No. of errors in 20,000 data symbols (64 QAM). DFE(N,2), f.r.=2.0 Hz, *SNR*=40.0 dB, $\frac{1}{\omega}$=1.237.
4	63
5	39
6	39
7	70
8	7,252

Table 7.4 Error count performance figures for revised SRK algorithm of [16] on Channel B with 64 QAM.

We can derive an approximate lower bound to ω below which instability arises. Recall from Appendix C and (7.23) that the i^{th} element of D_k is updated at each iteration as

$$d_1 = d_1 h_\omega \frac{(\eta+\hbar_{\mathrm{t}})}{(\alpha_1+\hbar_{\mathrm{t}})} \tag{7.25}$$

$$d_i = d_i h_\omega \beta_0 \kappa = d_i h_\omega \frac{(\alpha_{i-1}+\hbar_{\mathrm{t}})}{(\alpha_i+\hbar_{\mathrm{t}})} \quad \text{for} \quad 2\leq i\leq M \tag{7.26}$$

If the multiplying coefficient of d_i on the RHS of (7.25), (7.26) is always going to be >1, then d_i will keep on increasing, and therefore we have a situation of instability. If all the d_i, $1\leq i\leq M$, are infinitely large, we could expect all the α_i, $1\leq i\leq M$, also to be infinitely large, and thus the multiplying coefficient to be h_ω (>1) for all i, ensuring a condition of instability. Suppose that at any particular instant we have the element d_m infinitely large, where $1\leq m\leq M$, and the rest of the elements d_i, $i\neq m$, all finite. Then it may be assumed that all α_i, $1\leq i\leq m-1$, are finite, and that all α_i, $m\leq i\leq M$, are approximately equal and infinitely large. This would then make the multiplying coefficient in (7.25), (7.26) approximately $h_\omega-1$ for $i=m$, and h_ω for $i\neq m$. Further supposing that stability will be assured if the determinant of D_k, i.e. $\prod_{i=1}^{M} d_i$, keeps on decreasing, we then require for stability that

$$h_\omega^{M-1}(h_\omega-1) < 1 \qquad \Rightarrow \qquad M < 1+\frac{\ln(\frac{1}{\omega}-1)}{\ln\omega} \tag{7.27}$$

A graph of $1/\omega$ vs. M is plotted in fig. 7.10. For stability the combination of $1/\omega$ and M must produce a co-ordinate below the curve in fig. 7.10. When the adaptive DFE is implemented in the conventional manner, $M=N+g$. The predicted upper bounds to $1/\omega$ that give the threshold for instability compare reasonably well with the results in Table 7.4. For example, for DFE(10,2) the upper bound to $1/\omega$ from fig. 7.10 is approximately 1.173, and in Table 7.4 there is a large change in error count going from $1/\omega=1.175$ to 1.200. The state of instability may actually already exist even though a low error count has been recorded, since the detection process only fails when any of the $\{d_i\}$ reach the maximum positive number capable of being stored on the computer. This can be seen in the results for DFE(15,2), the first column listing counts taken from 40,000 symbols and suggesting a threshold for instability between

$1/\omega$=1.125 and 1.150, while the second column lists counts taken from half the number of symbols (therefore detection process has not been running for so long) and suggests a threshold for instability between $1/\omega$=1.150 and 1.175; the predicted threshold for $1/\omega$ is approximately 1.134.

It was observed for the DFE(6,2) with 64 QAM on Channel B, at a 2.0 Hz fade rate, that the probability of error can barely be reduced below 10^{-3} without causing instability by further reduction in ω. Apart from producing the results in this section, the revised SRK algorithm of [16] is not used further in this thesis.

7.4 Performance of DFE(6,g)-II$_1$

Unless stated otherwise, in the rest of this thesis the DFE should be assumed to be implemented by way of the CE method.

Figs. 7.11–7.14 depict the long-term performance of the DFE(6,g)-II$_1$. Except for the mode I$_1$ curves, which are shown for ease of comparison with mode II$_1$, the continuous curves represent the performance offered by the SRK algorithm while the plot symbols represent that from the SD algorithm, both algorithms having ω and μ' respectively set to their optimum theoretical values as detailed in chapter 6. As usual, in measuring long-term performance there is a warm-up time of 1.67 fade periods prior to transmission over the fading patterns, to remove transient start-up effects.

Figs. 7.11 and 7.12, which refer to Channel A, indicate that there is very little difference between the SRK and SD algorithms, up to a fade rate of 2.0 Hz. In figs. 7.13 and 7.14, which refer to Channel B, there is a more noticeable degradation of the SD algorithm with respect to the SRK algorithm, this being apparent at the high *SNR* end of the curves for 16 and 64 QAM.

The number of computations needed by the SD algorithm is $10(g+1)$ real multiplications and $8(g+1)$ real additions, and its use in the CE method thus brings about a reduction in both multiplications and additions of approximately 6–7% for DFE(6,1) and 13% for DFE(6,2).

Clearly, in view of the performance offered by the SD algorithm, its use is worth considering if computational complexity is critical. However, since the SRK algorithm generally has an advantage (albeit small at times) over the SD algorithm, both in terms of long-term performance and start-up convergence, its use will be more predominant than the latter's for the remaining work in this thesis. The discussion in the rest of this section is centred on results from the SRK algorithm.

The curves for 4 QAM in figs. 7.11–7.14 indicate that there is little degradation in going from mode I_1 to mode II_1; at $P_e=10^{-6}$ the degradation, in terms of SNR, is about 1.0 dB and 1.5 dB on Channel A and Channel B respectively. The difference between 1.0 Hz and 2.0 Hz fade rates for 4 QAM appears to be very small in the P_e curves, the difference being slightly more marked in the ξ curves. Overall the results for 4 QAM suggest that fade rates higher than 2.0 Hz can be coped with quite easily. The degradation from mode I_1 to mode II_1 is larger for 16 QAM, and more so for 64 QAM, this being a consequence of having to operate at increasingly higher values of SNR. For example, observe the figures in Table 7.5 for the SNR degradation from mode I_1 to mode II_1, the information being taken from figs. 7.11 and 7.13. For 64 QAM, it looks as though the required SNR would have to be in excess of 50.0 dB in mode II_1 in order to achieve a P_e of 10^{-6} on Channel A, and on Channel B at f.r.=2.0 Hz.

		f.r.=1.0 Hz		f.r.=2.0 Hz	
		$P_e=10^{-3}$	$P_e=10^{-6}$	$P_e=10^{-3}$	$P_e=10^{-6}$
Channel A	16 QAM	1.1	2.0	1.5	3.3
	64 QAM	1.5	—	2.7	—
Channel B	16 QAM	1.2	2.6	1.9	3.8
	64 QAM	2.1	4.6	3.8	—

Table 7.5 SNR degradation (dB) from mode I_1 to mode II_1.

The degradation from mode I_1 to mode II_1 is generally greater on Channel B than on Channel A, because B's longer channel response vector can be known less accurately. The extra path component of Channel B, however, ensures that its mode II_1 P_e performance is usually still much better than Channel A's, in contrast to its ξ performance, which is worse than Channel A's. We saw in chapter 5 that the *SNR* difference between Channel A and Channel B is much smaller for the ξ criterion than for the P_e criterion, indicating that the ξ criterion does not highlight the diversity advantage of Channel B.

At time kT the receiver assumes the channel vector is equal to the estimated vector $\underline{c}_{k-1}$. To the receiver, the tracking algorithm error $\epsilon_k = r_k - \underline{x}_k^t \underline{c}_{k-1}$ is the sole corrupting influence on the information bearing part of the received sample r_k. The mean square value of ϵ_k, where the expectation is over the gaussian noise and data symbols, varies with k because of the time variation of the channel. Thus we may think of the interference in r_k as being from a zero mean noise quantity, not necessarily gaussian, that has a time-varying variance. The ratio of σ_s^2 to the mean square value of ϵ_k can therefore be regarded as an effective signal-to-noise ratio at time kT. Averaging the mean square value of ϵ_k over the fading statistics of the channel gives us ϵ (6.1), and the ratio σ_s^2/ϵ is thus a kind of average signal-to-noise ratio that relates to long-term performance over the channel. The quantity $10\log_{10}[\epsilon/(\rho\sigma_s^2)]$ represents the dB loss with respect to the signal-to-noise ratio $1/\rho$. As *SNR* increases, then $\epsilon/(\rho\sigma_s^2)$ gets larger, as can be seen in figs. 6.2 and 6.3 of chapter 6, and thus we may expect the dB loss from mode I_1 to mode II_1 to get larger as well.

Assuming that the estimated vector $\underline{c}_{k-1}$, in the absence of decision errors, follows the same basic time variation as does the channel vector, consider now a scheme in which the performance level in mode II_1, at a given value of *SNR*, is predicted as being the performance level in mode I_1 at an *SNR* value of σ_s^2/ϵ, where ϵ is obtained from mode II_1. Using figs. 6.2 and 6.3 and the mode I_1 performance curves we thus form the dashed curves in figs. 7.11–7.14. The prediction of the ξ performance is extremely good for both channels. The prediction of the P_e performance tends to become worse at the lower values of P_e, this appearing to be more so for Channel A than Channel B. This could be due to the fact that the P_e measurement depends to some extent on the fourth order statistics of ϵ_k, and also that for low values it tends to be less indicative of truely long-term behaviour, a condition that is worse for Channel A, as was discussed in chapter 5.

Note that the feedforward section of the DFE(N,g) at time kT contains N received samples r_{k+i}, $0 \leq i \leq N-1$. If the DFE taps are determined from the estimated vector $\underline{c}_{k-1}$, then each received sample in the feedforward section can be imagined as having an effective "noise" given by $r_{k+i} - \underline{x}^{t}_{k+i}\underline{c}_{k-1}$, this quantity having an expected square value (over gaussian noise, data symbols and fading channel statistics) that increases with i. In the prediction scheme described above we are effectively ignoring the variation in "noise" variance over the received samples in the DFE.

On the whole the prediction of the P_e and ξ performance in mode II$_1$, based on the performance in mode I$_1$, appears reasonably good given the simplistic nature of the method. In a similar fashion the performance by way of the SD algorithm can be predicted too. Given that we have mode I$_1$ performance curves, it is much easier to produce curves of ϵ/σ_s^2 vs. SNR than it is to simulate a DFE-II$_1$ and measure P_e, ξ. One could alternatively use the theoretical expressions for ϵ derived in chapter 6 for both the SRK and SD algorithms, thereby reducing further the need for simulations, though the prediction accuracy for the SRK algorithm, unlike for the SD, is likely to suffer at high SNR where the theoretical estimate of ϵ is not so good.

7.5 MLSE/near-MLSE and DFE(6,g)

Figs. 7.15 and 7.16 depict the long-term P_e(E.C.) vs. SNR perfomance of the MLSE/near-MLSE and DFE(6,g) in mode II, at a fade rate of 1.0 Hz. The E.C. curves consist of straight lines between the data points. Tables 7.6 and 7.7 give the number of errors occurring in the E.C. measurement runs. The tap values of the DFE, and therefore also the pre-filter taps for the near-MLSE, are determined according to the CE method with the channel estimate being produced by the SRK algorithm. The warm-up time is 1.67 fade periods prior to measurement. The channel estimate used by the MLSE/near-MLSE is the same as that for the DFE, so in effect we are ignoring any limitation due to the delay in detection of the

SNR	Errors in 60,000, f.r.=1.0 Hz			Errors in 30,000, f.r.=2.0 Hz		
(dB)	DFE(6,1)		MLSE/	DFE(6,1)		MLSE/
4 QAM	Mode II$_1$	Mode II$_2$	near-MLSE	Mode II$_1$	Mode II$_2$	near-MLSE
7.5	4,843 (4,657)	5,986 (5,730)	5,119 (4,788)	2,507	3,127	2,742
10.0	2,345 (2,139)	3,028 (2,768)	2,411 (2,219)	1,170	1,514	1,200
12.5	1,028 (912)	1,316 (1,186)	998 (931)	538	726	533
15.0	457 (400)	603 (548)	468 (416)	223	280	219
20.0	101 (92)	154 (136)	93 (78)	59	76	46
24.0	26 (20)	29 (21)	25 (18)	10	9	9
27.5	6 (4)	6 (5)	6 (6)	2	5	3
30.0	1 (1)	3 (1)	2 (2)	0	0	0
16 QAM						
15.0	7,588 (6,905)	11,276 (10,379)	9,912 (8,998)	4,010	5,958	5,323
20.0	1,641 (1,429)	2,993 (2,649)	2,286 (1,880)	902	1,567	1,242
25.0	343 (264)	615 (458)	464 (405)	184	329	262
30.0	88 (61)	175 (160)	127 (89)	40	71	51
32.5	36 (22)	77 (51)	54 (24)	16	30	15
35.0	8 (8)	16 (16)	12 (10)	7	22	10
37.5	7 (2)	13 (7)	7 (7)	3	19	8
40.0	1 (0)	6 (0)	6 (0)	0	0	0
64 QAM						
25.0	3,374 (2,711)	7,964 (7,017)	6,385 (5,485)	1,893	4,668	3,865
30.0	695 (507)	1,804 (1,245)	1,211 (921)	426	1,077	669
35.0	180 (120)	472 (311)	345 (249)	104	260	199
37.5	85 (54)	284 (220)	189 (157)	52	146	113
40.0	42 (18)	176 (77)	120 (27)	29	109	74
42.5	19 (9)	128 (76)	72 (17)	15	79	40
45.0	11 (1)	100 (9)	59 (0)	7	43	38

Table 7.6 Error counts for mode II on Channel A, using SRK algorithm. Figures in (.) indicate values obtained for mode I (chapter 5).

SNR	Errors in 60,000, f.r.=1.0 Hz			Errors in 30,000, f.r.=2.0 Hz		
(dB)	DFE(6,2)		MLSE/	DFE(6,2)		MLSE/
4 QAM	Mode II_1	Mode II_2	near-MLSE	Mode II_1	Mode II_2	near-MLSE
7.5	5,060 (4,647)	7,164 (6,609)	5,600 (5,154)	2,577	3,674	2,806
12.5	919 (800)	1,497 (1,316)	890 (748)	483	795	487
16.5	146 (109)	274 (208)	133 (107)	76	153	53
20.0	24 (22)	35 (34)	20 (16)	18	34	13
22.5	9 (5)	12 (8)	4 (3)	4	10	2
16 QAM						
17.5	4,199 (3,558)	9,422 (8,340)	7,365 (6,097)	2,292	4,933	3,964
22.5	610 (420)	2,061 (1,486)	1,222 (908)	366	1,208	754
26.5	85 (44)	369 (101)	126 (59)	65	204	127
30.0	13 (6)	57 (16)	40 (3)	11	48	31
31.0	5 (3)	32 (3)	6 (0)	7	39	13
32.5	0 (0)	0 (0)	0 (0)	2	13	13
64 QAM						
25.0	4,267 (3,113)	16,602 (13,872)	11,619 (9,250)	2,518	8,977	6,857
30.0	673 (338)	6,444 (3,812)	2,255 (1,222)	450	3,598	1,514
32.5	194 (76)	1,863 (736)	427 (160)	185	2,270	538
35.0	57 (22)	505 (77)	130 (45)	72	910	138
37.5	13 (2)	103 (74)	69 (0)	29	840	64
40.0	2 (0)	22 (0)	8 (0)	11	61	39

Table 7.7 Error counts for mode II on Channel B, using SRK algorithm. Figures in (.) indicate values obtained for mode I (chapter 5).

MLSE/near-MLSE. In this way the two detection devices are on an equal footing when it comes to accuracy in channel estimation. Figs. 7.15, 7.16 and Tables 7.6, 7.7 also show the performance in mode I for ease of comparison with mode II. Also shown in Tables 7.6 and 7.7 are mode II error counts for runs at a 2.0 Hz fade rate, the counts now being over only 30,000 transmitted symbols, these symbols being the first 30,000 of the 60,000 symbols used at the fade rate of 1.0 Hz. The error counts for 2.0 Hz when doubled are, as expected, generally greater than the corresponding counts for 1.0 Hz. The data for 2.0 Hz is not plotted in figs. 7.15 and 7.16 so as not to over-complicate the graphs, and in any case, the essential message the data conveys, which will be discussed shortly, can easily be seen from the tables. The performance with the SD algorithm is not very different from that with the SRK algorithm, and so is not shown.

As we saw in the last section, the DFE-I_1 and DFE-II_1 produce lower error rates, at a given *SNR*, on Channel B than on Channel A. With 16 and 64 QAM in particular, however, the near-MLSE and DFE-I_2,II_2 produce lower E.C. error rates on Channel A than on Channel B above about $P_e = 10^{-2}$. This is probably because of the more dispersive nature of Channel B, producing longer bursts of errors. The extra path, or order of diversity, in Channel B ensures that when $P_e(\text{E.C.}) \lesssim 10^{-2}$, the P_e's produced by the near-MLSE and DFE-I_2,II_2 on this channel are lower, at a given *SNR*, than on Channel A.

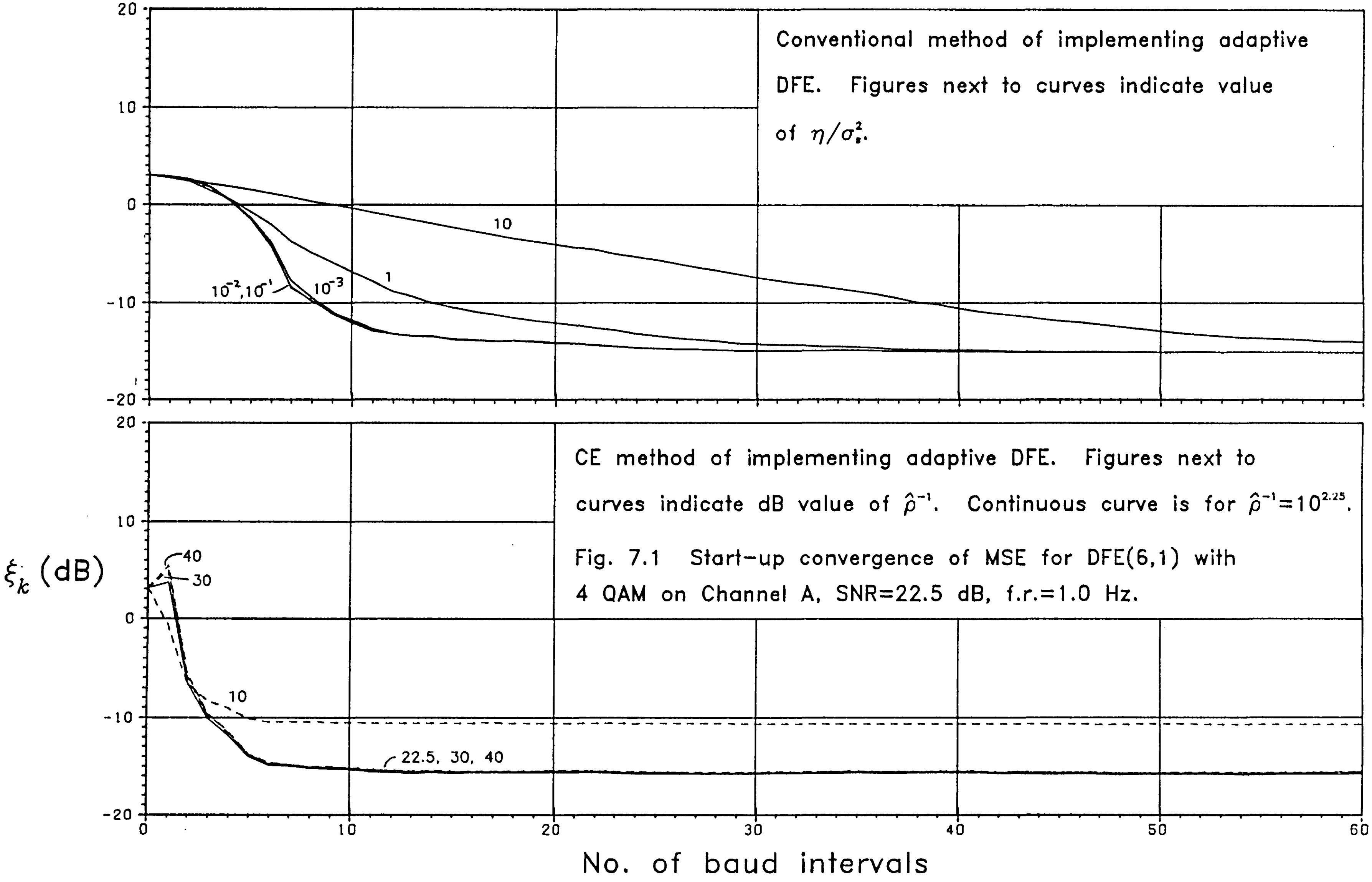
The results for mode II reaffirm most of the observations made previously in chapter 5 for mode I. For 4 QAM the degradation caused by propagation in the DFE is fairly small, and the MLSE is quite close in performance to the DFE-II_1. As the constellation size increases to 16 and 64, the effect of propagation gets progressively worse, and is particularly severe for 64 QAM, as can be seen from the figures in Tables 7.6 and 7.7. Though the MLSE performs as well as the DFE-II_1 for 4 QAM, the near-MLSE performs noticeably worse than the DFE-II_1 for 16 QAM, with the gap even wider for 64 QAM. It is interesting to note that for both channels the gap in performance between the near-MLSE and DFE in mode II_1 generally appears larger than the coresponding gap in mode I_1. For example, observe in fig. 7.16 for

Channel B that in mode I_1 the near-MLSE can, at high enough *SNR*, produce a slightly lower error count than the DFE for both 16 and 64 QAM, but definately not so when in mode II_1. This would suggest that the near-MLSE degrades more than the DFE (in the absence of feedback errors) when both devices are subject to the same error in the channel estimate. On the other hand, the near-MLSE does perform better than the DFE-II_2, which leads us to conclude that in the situations studied here the MLSE/near-MLSE is superior to the DFE only because of the latter device's inclination to propagate errors. From figs. 7.15 and 7.16 at the *SNR* for which $P_e = 10^{-3}$ for the MLSE/near-MLSE in mode II, the corresponding P_e for the DFE-II_2 is, at most, just slightly less than 2×10^{-3}.

As far as computational complexity is concerned, with the values of g used here the MLSE for 4 QAM is only marginally more complex than the DFE. With the near-MLSE, however, the difference lies in that the DFE uses a feedback filter and a simple threshold detector at the output of the feedforward section, whereas the near-MLSE, with the feedforward section now acting as its pre-filter, uses the detection algorithm detailed in Appendix G (and given as a Fortran program in Appendix I). The additional cost over the DFE of the near-MLSE is thus considerable. As a practical example of the additional cost involved, a complete simulation run of 60,000 symbols on the computer (includes everything from generation of data symbols, fading channel, etc. to update of SRK algorithm, receiver devices and the measurement processes themselves, but excluding the analytical P_e estimate) takes approximately 4–5 times longer with the near-MLSE than it does with the DFE. The performance of the near-MLSE could be improved if it were allowed to operate with more than 16 and 64 survivors and expanded sequences respectively [42], but probably not without a significant increase in complexity.

Although the MLSE/near-MLSE has used exactly the same channel estimate as the DFE in producing the results of this section, in practice unless some type of early detection, such as for example a DFE operating in parallel, or some type of extrapolation technique on the channel estimate (see section 6.4) is employed, the delay in detection of the MLSE/near-MLSE can lead to a significant deterioration in the accuracy of the channel estimate. Another point to bear in mind is from the work in [49], which shows that in the presence of phase jitter, on an otherwise time-invariant channel, the performance of the VA can degrade to a greater extent than that of the DFE. This therefore means there has to be greater emphasis on accuracy in any carrier phase recovery scheme if an MLSE/near-MLSE is employed. Assuming the carrier phase recovery scheme is data-aided, as was described in chapter 4, it must therefore also confront the problem of the delay in detection of the MLSE/near-MLSE. One could view the presence of phase jitter as simply being a source of error in the receiver's knowledge of the channel response vector, and thus the results of [49] may offer some support to our earlier remark about the near-MLSE degrading more than the DFE when there is error in the channel estimate. We can reasonably conclude that the performance of the MLSE/near-MLSE depicted so far is the best that can be expected with respect to the DFE.

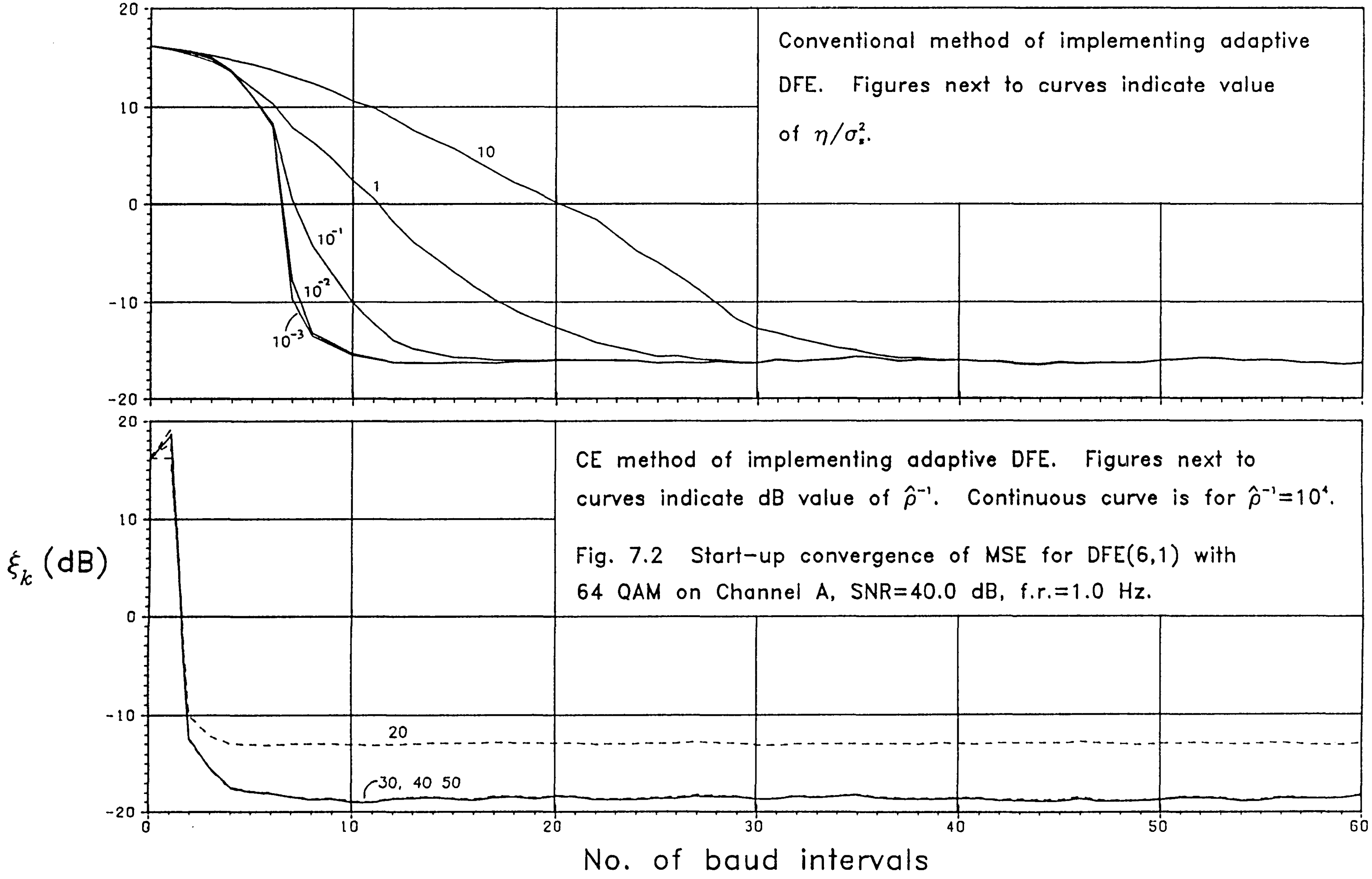
On a final note we mention a near-MLSE developed in [18] for 16 QAM, which operates without any digital pre-filter, requiring only an estimate of the channel vector. The channel estimate must be examined regularly to determine the most "significant" component. The scheme is quite different from any near-MLSE based on the concept of a reduced-state VA [18]. The motive behind the strategy adopted in [18] lies in the assertion that the adaptive pre-filter is difficult to hold correctly adjusted, compared to estimation of the channel vector which can be achieved more rapidly and accurately. This claim is probably made under the assumption that the pre-filter is implemented by way of the conventional method, using the SD algorithm. For time-varying channels like the HF, the conventional RLS implementation of the adaptive DFE is vastly superior in performance to the conventional implementation using the SD algorithm [15], [64]. We have shown that the CE method of implementing the adaptive DFE is superior in performance to the conventional RLS method. In operating without a pre-filter, the near-MLSE scheme of [18] is more complex [97] than similar-performing near-MLSE schemes that use a pre-filter (assuming it is correctly adjusted), such as the ones in [42].



Conventional method of implementing adaptive DFE. Figures next to curves indicate value of η/σ_s^2.

CE method of implementing adaptive DFE. Figures next to curves indicate dB value of $\hat{\rho}^{-1}$. Continuous curve is for $\hat{\rho}^{-1}=10^{2.25}$.

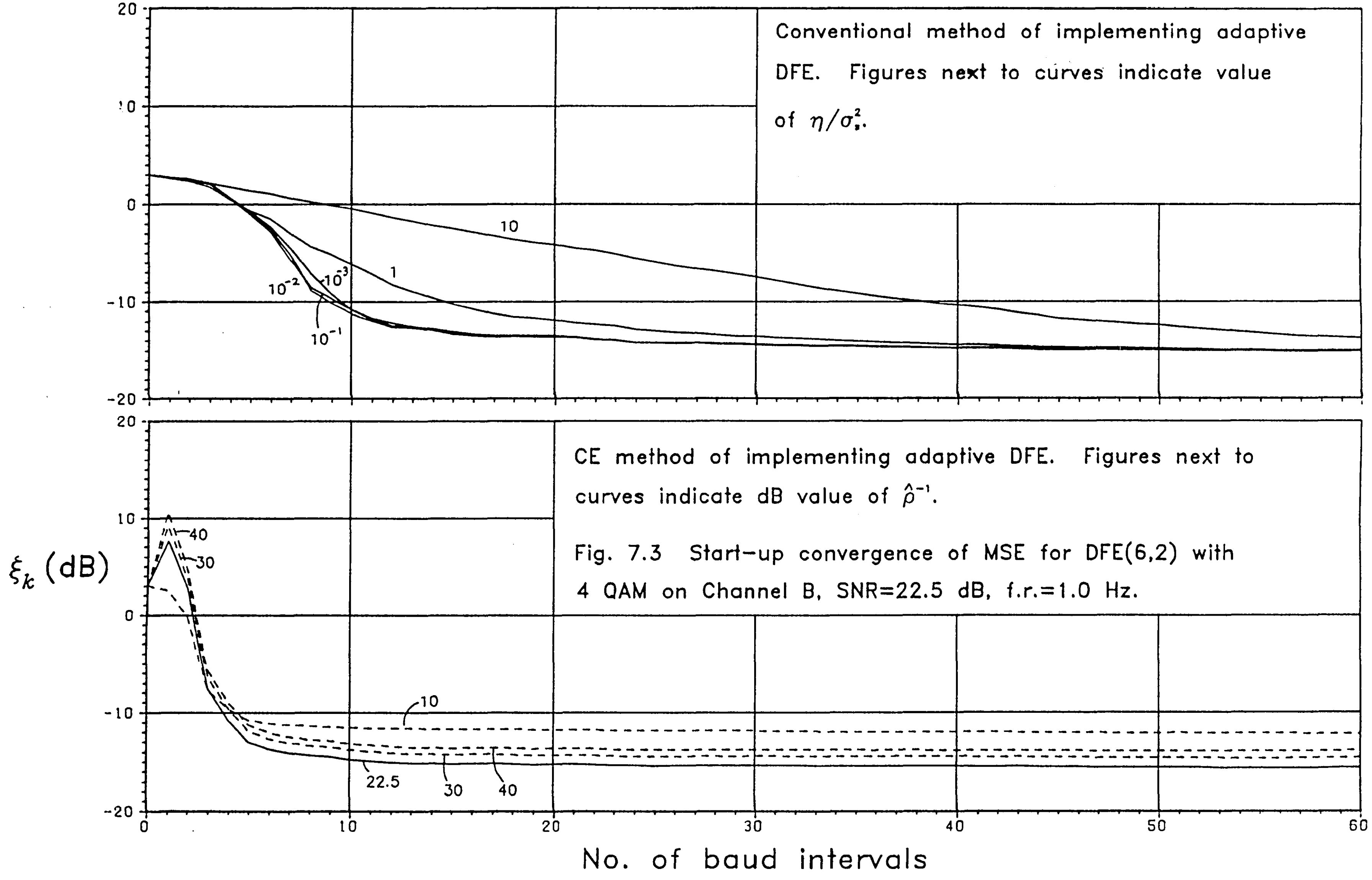
Fig. 7.1 Start-up convergence of MSE for DFE(6,1) with 4 QAM on Channel A, SNR=22.5 dB, f.r.=1.0 Hz.



Conventional method of implementing adaptive DFE. Figures next to curves indicate value of η/σ_s^2.

CE method of implementing adaptive DFE. Figures next to curves indicate dB value of $\hat{\rho}^{-1}$. Continuous curve is for $\hat{\rho}^{-1}=10^4$.

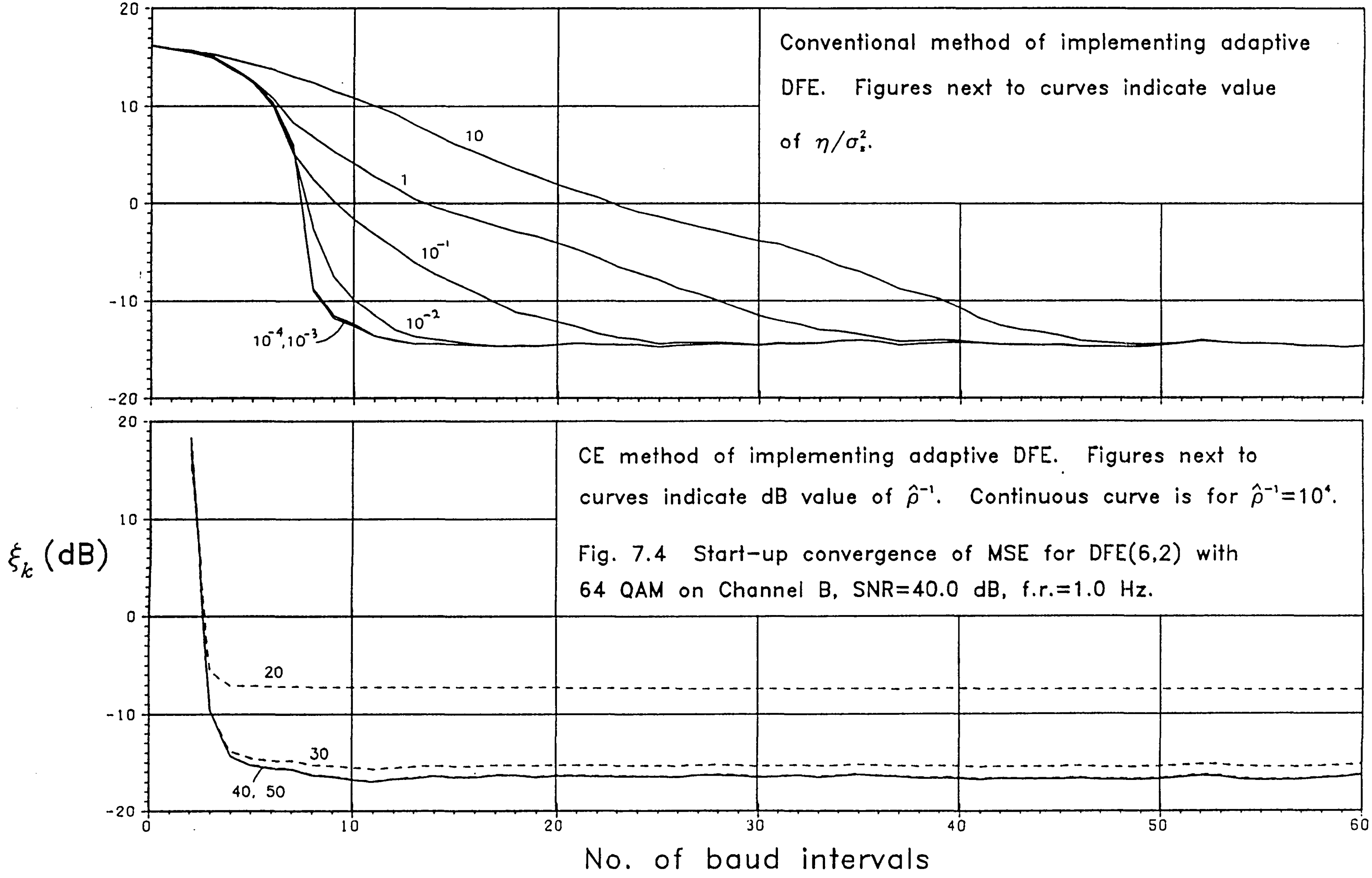
Fig. 7.2 Start-up convergence of MSE for DFE(6,1) with 64 QAM on Channel A, SNR=40.0 dB, f.r.=1.0 Hz.



Conventional method of implementing adaptive DFE. Figures next to curves indicate value of η/σ_s^2.

CE method of implementing adaptive DFE. Figures next to curves indicate dB value of $\hat{\rho}^{-1}$.

Fig. 7.3 Start-up convergence of MSE for DFE(6,2) with 4 QAM on Channel B, SNR=22.5 dB, f.r.=1.0 Hz.



Conventional method of implementing adaptive DFE. Figures next to curves indicate value of η/σ_s^2.

CE method of implementing adaptive DFE. Figures next to curves indicate dB value of $\hat{\rho}^{-1}$. Continuous curve is for $\hat{\rho}^{-1}=10^4$.

Fig. 7.4 Start-up convergence of MSE for DFE(6,2) with 64 QAM on Channel B, SNR=40.0 dB, f.r.=1.0 Hz.

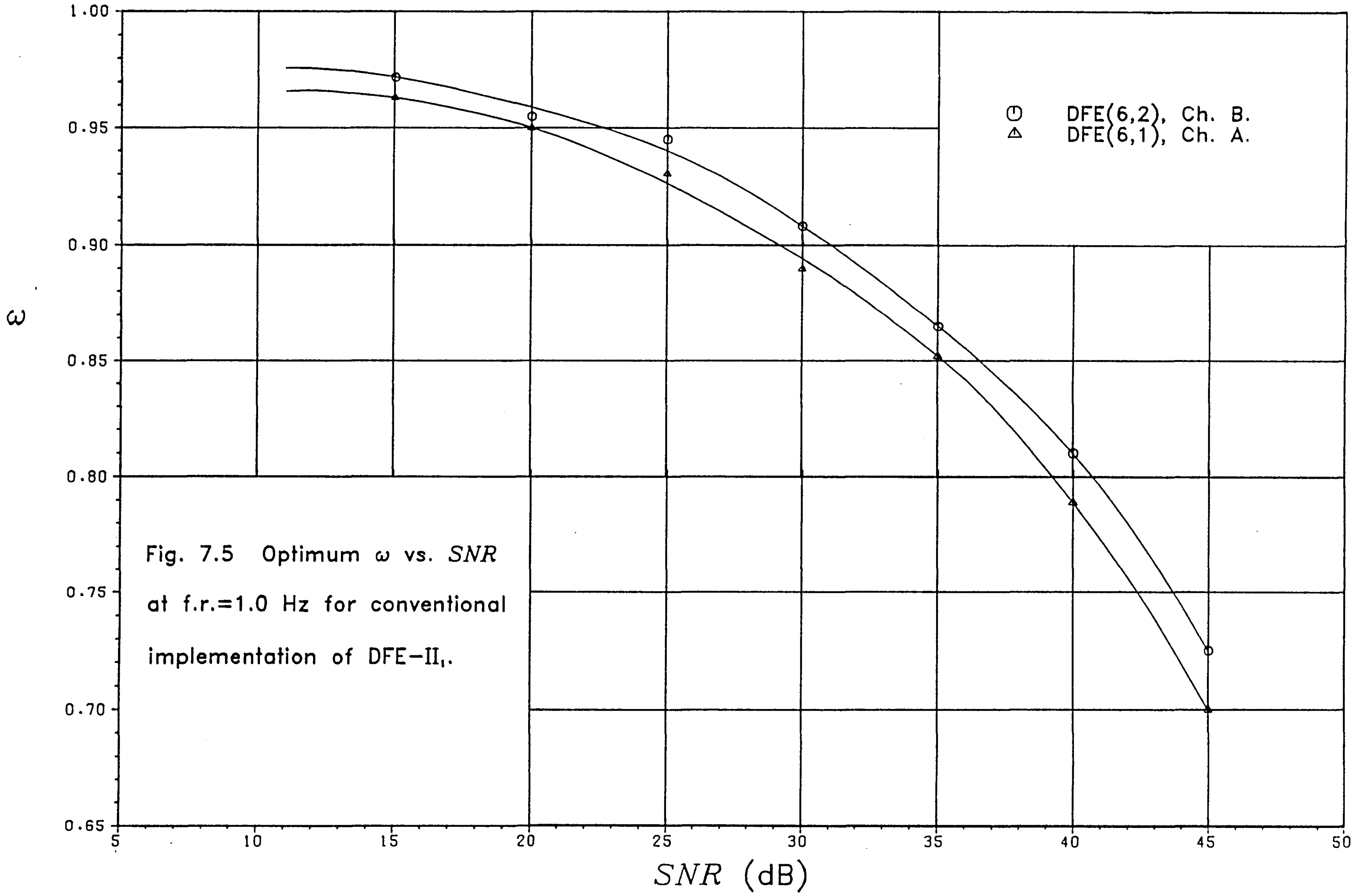


Fig. 7.5 Optimum ω vs. *SNR* at f.r.=1.0 Hz for conventional implementation of DFE-II$_1$.

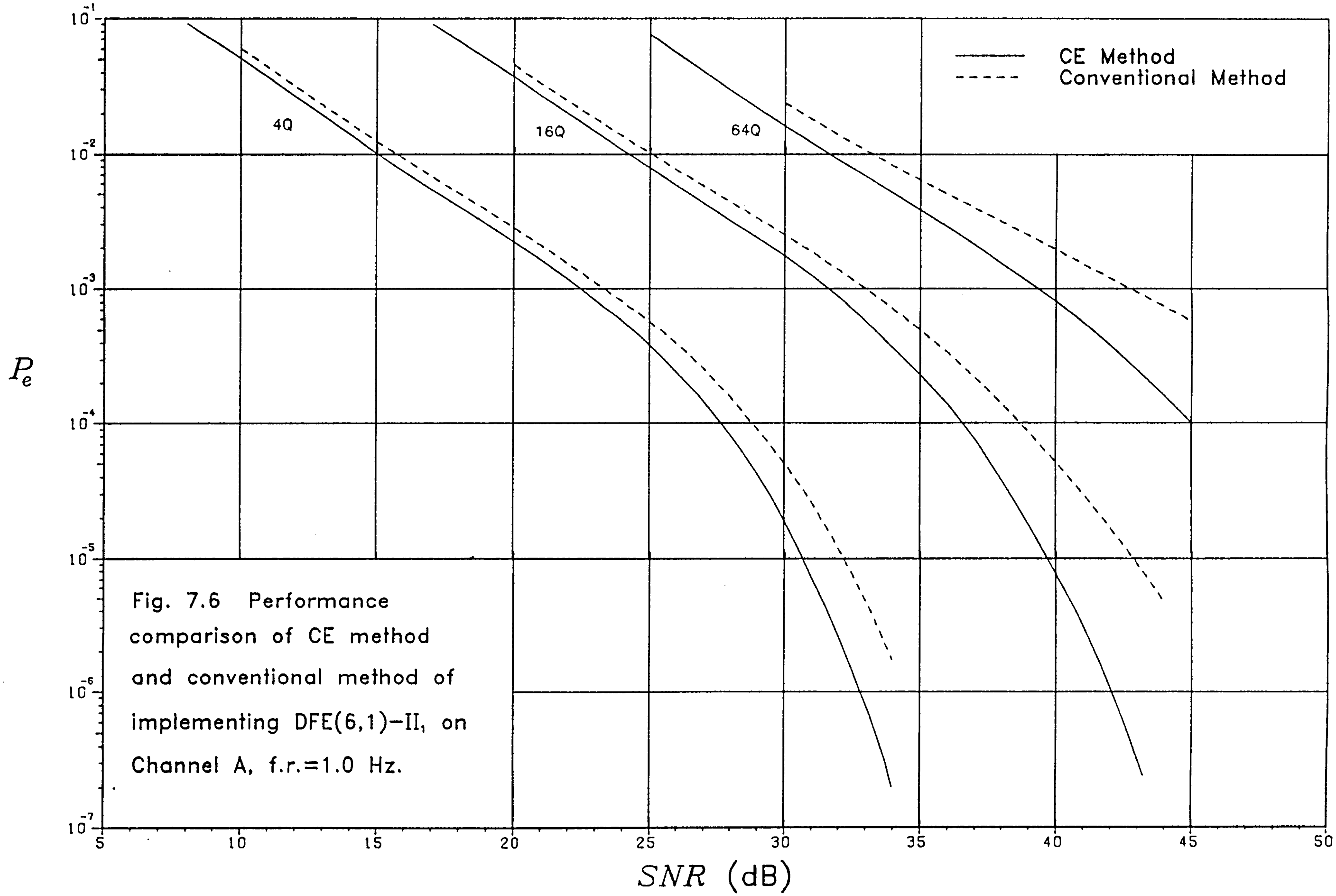


Fig. 7.6 Performance comparison of CE method and conventional method of implementing DFE(6,1)-II$_1$ on Channel A, f.r.=1.0 Hz.

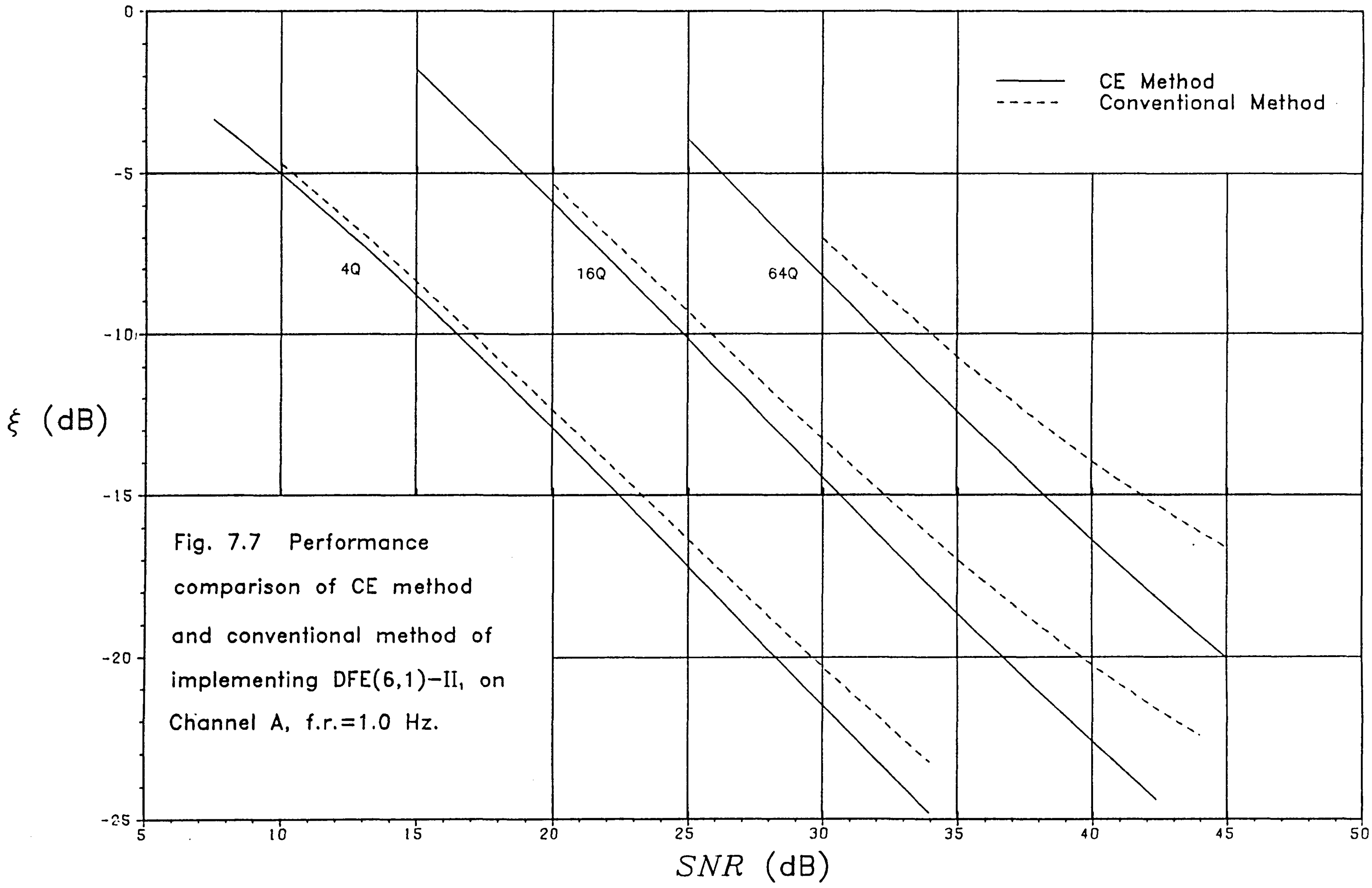


Fig. 7.7 Performance comparison of CE method and conventional method of implementing DFE(6,1)–II, on Channel A, f.r.=1.0 Hz.

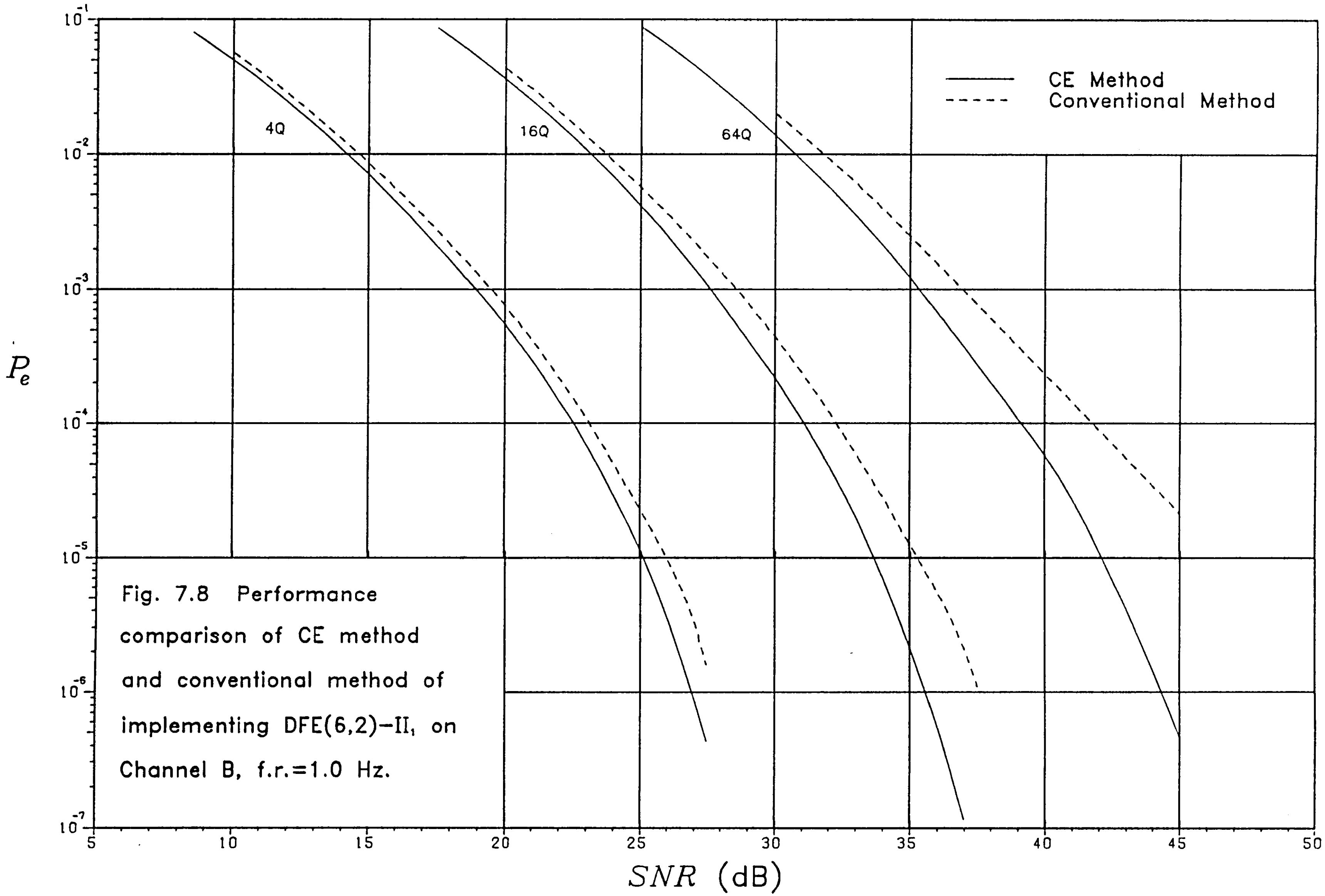


Fig. 7.8 Performance comparison of CE method and conventional method of implementing DFE(6,2)–II$_1$ on Channel B, f.r.=1.0 Hz.

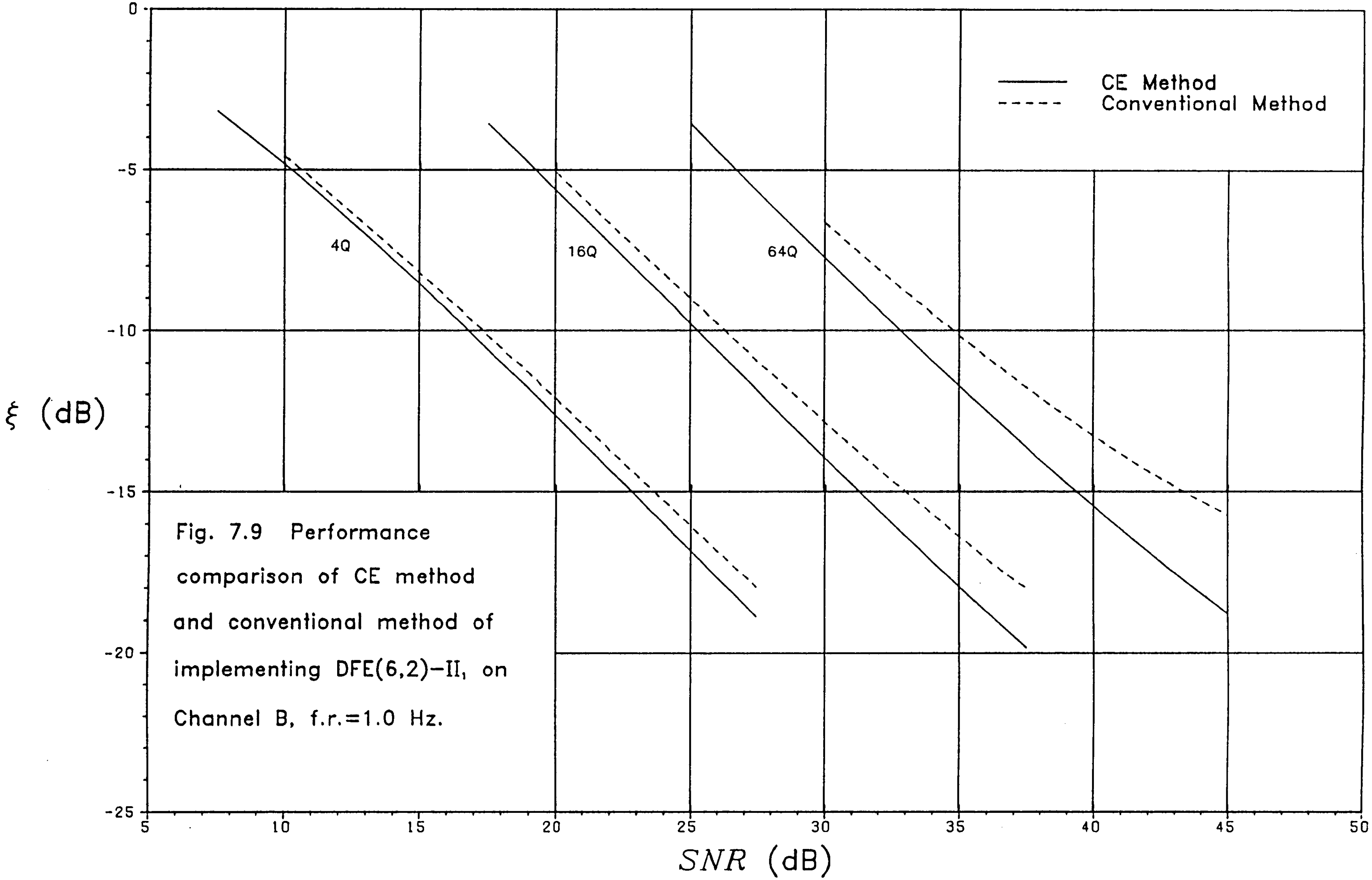


Fig. 7.9 Performance comparison of CE method and conventional method of implementing DFE(6,2)–II$_1$ on Channel B, f.r.=1.0 Hz.

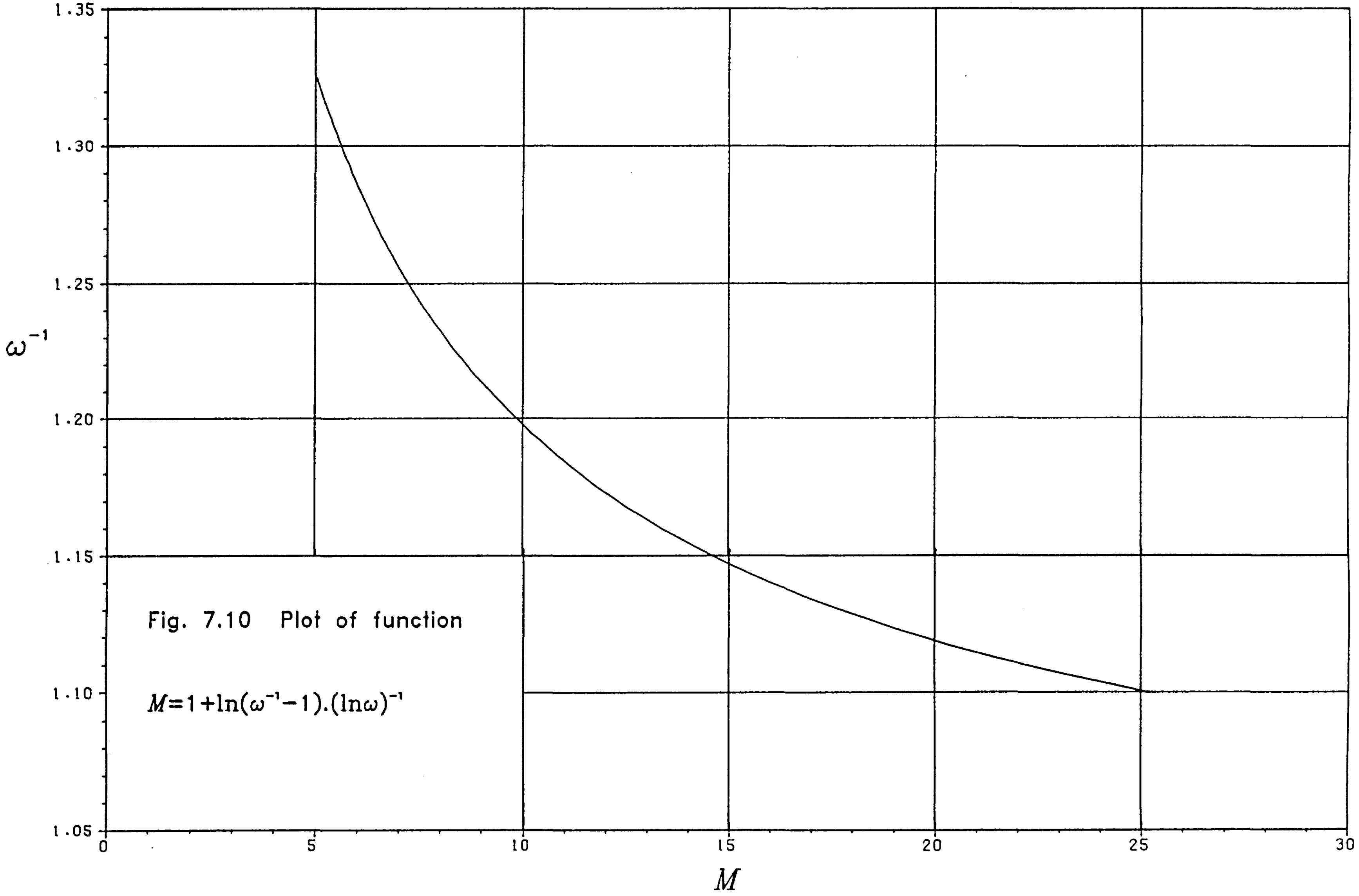


Fig. 7.10 Plot of function

$M=1+\ln(\omega^{-1}-1).(\ln\omega)^{-1}$

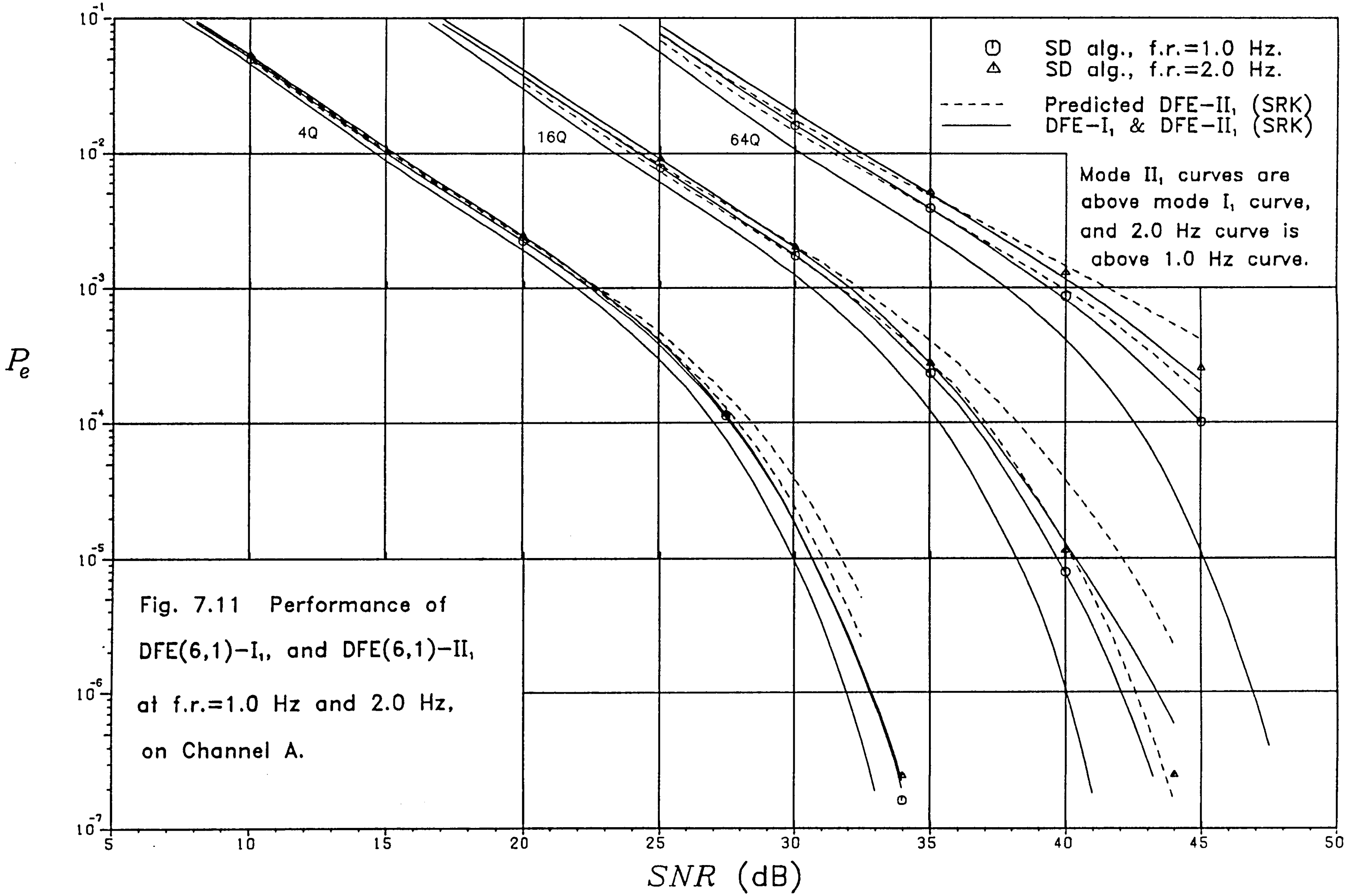
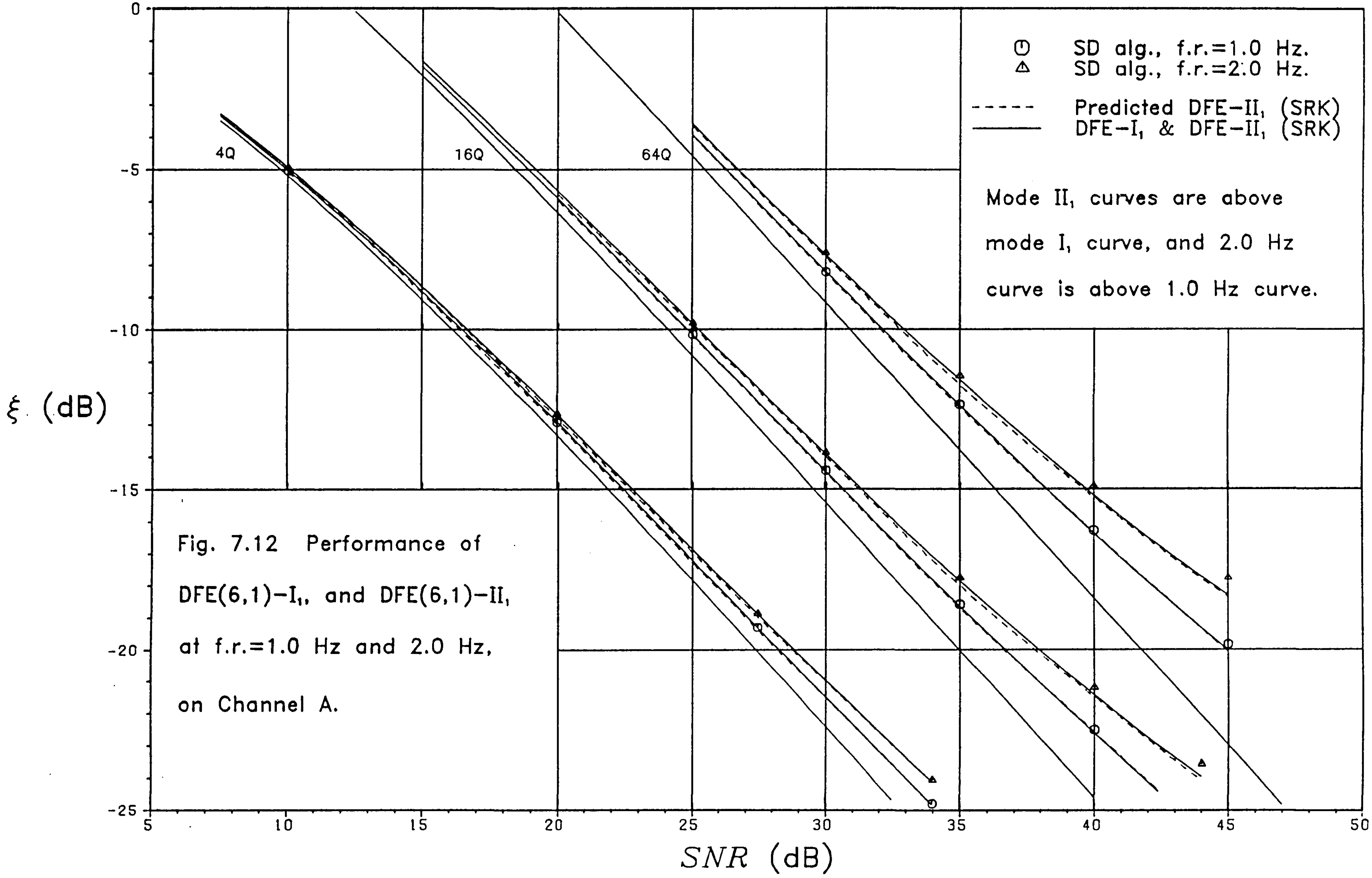


Fig. 7.11 Performance of DFE(6,1)-I$_1$, and DFE(6,1)-II$_1$ at f.r.=1.0 Hz and 2.0 Hz, on Channel A.



Mode II_1 curves are above mode I_1 curve, and 2.0 Hz curve is above 1.0 Hz curve.

Fig. 7.12 Performance of $\text{DFE}(6,1)\text{-I}_1$, and $\text{DFE}(6,1)\text{-II}_1$ at f.r.=1.0 Hz and 2.0 Hz, on Channel A.

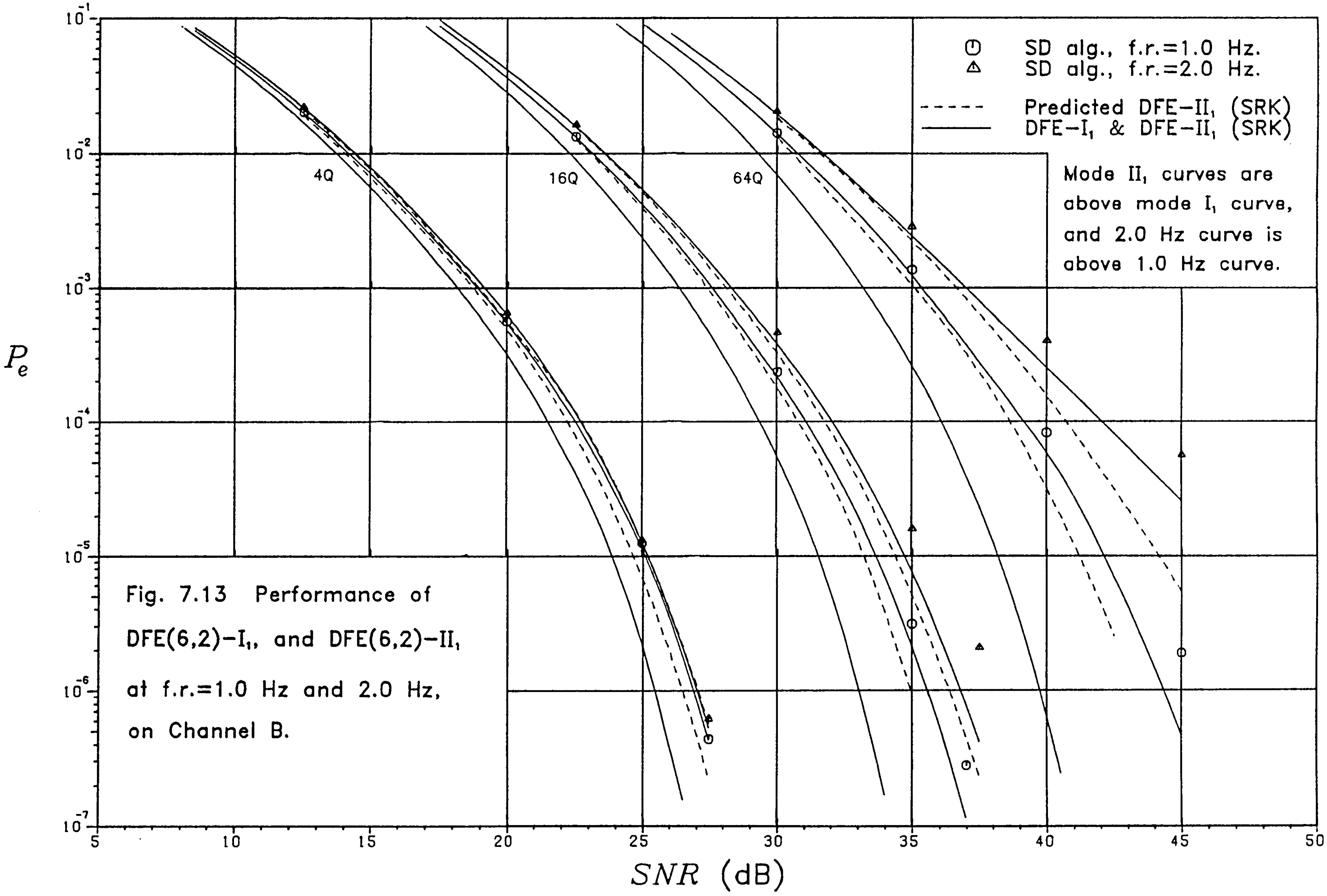


Fig. 7.13 Performance of DFE(6,2)-I₁, and DFE(6,2)-II₁ at f.r.=1.0 Hz and 2.0 Hz, on Channel B.

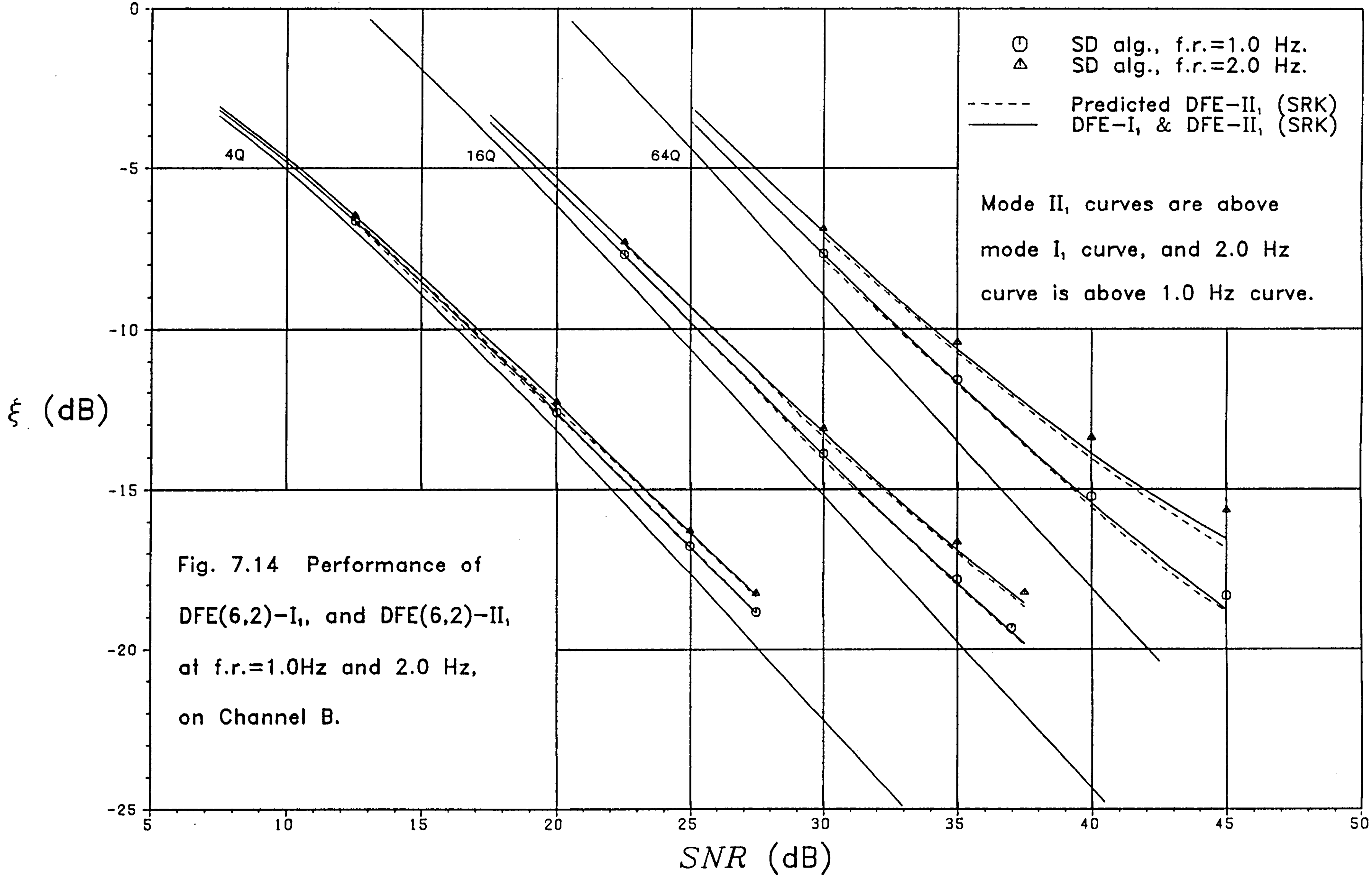


Fig. 7.14 Performance of DFE(6,2)−I₁, and DFE(6,2)−II₁ at f.r.=1.0Hz and 2.0 Hz, on Channel B.

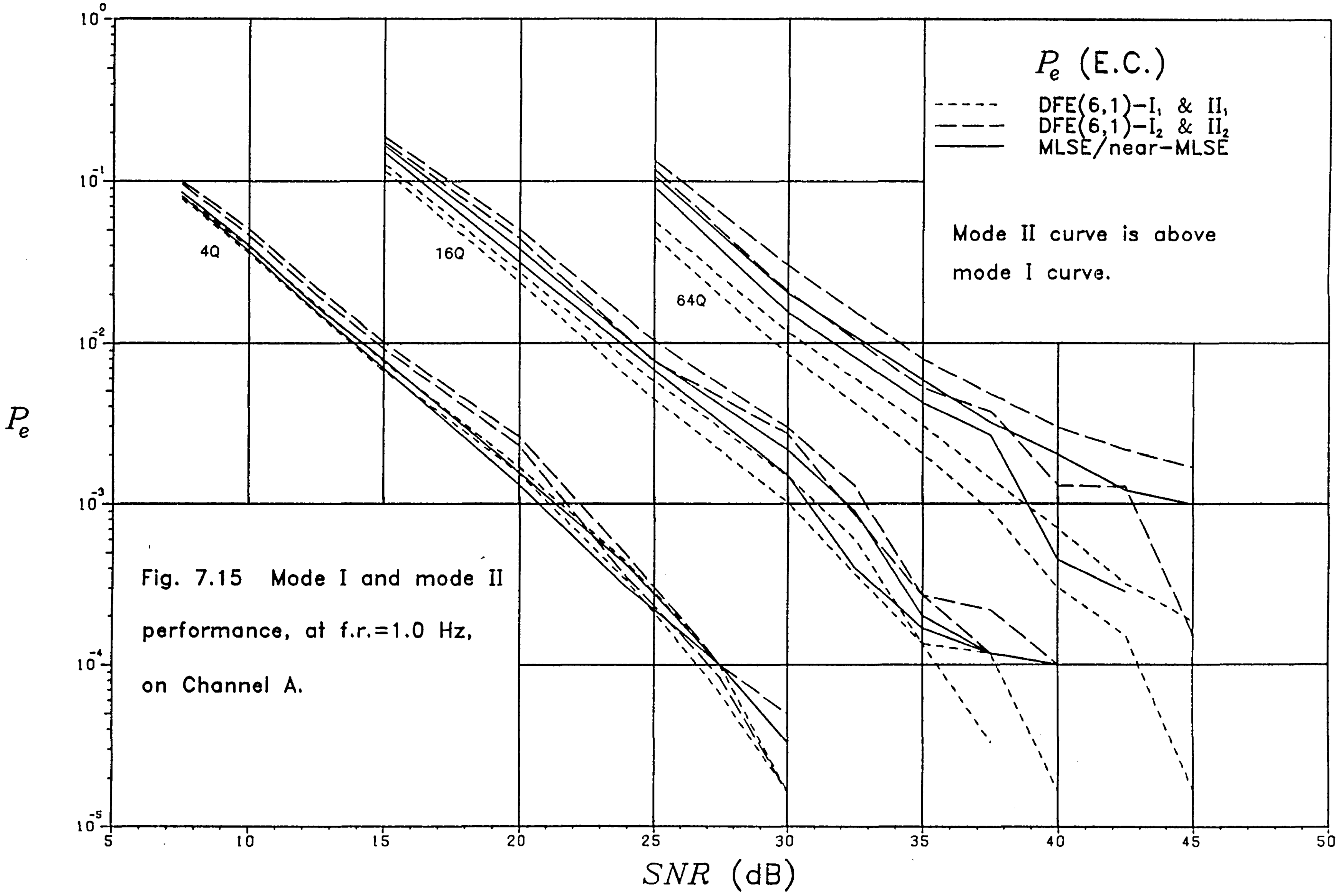


Fig. 7.15 Mode I and mode II performance, at f.r.=1.0 Hz, on Channel A.

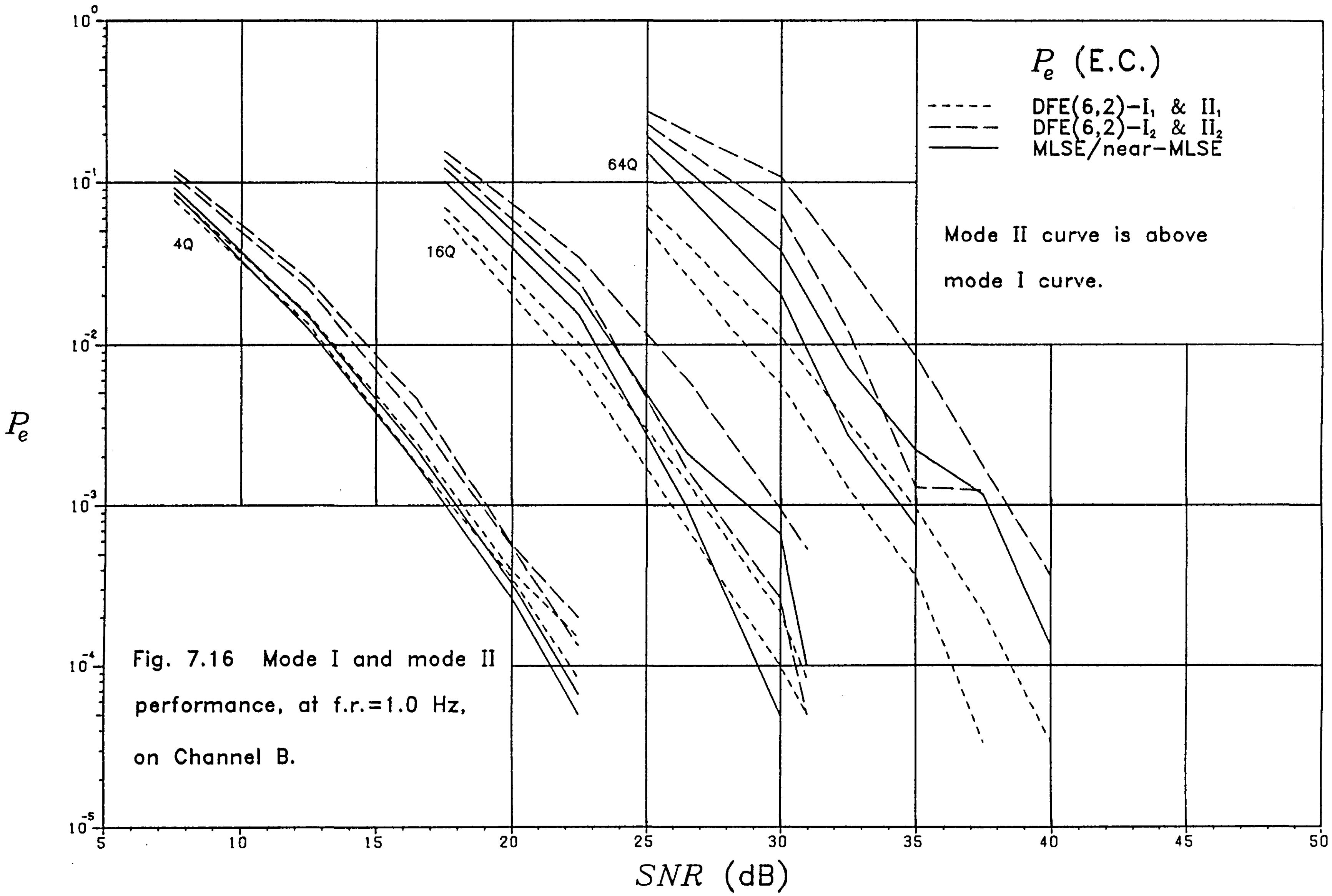


Fig. 7.16 Mode I and mode II performance, at f.r.=1.0 Hz, on Channel B.

Chapter 8
MODE III PERFORMANCE

The performance of the receiver up to now has been observed under condition that the tracking algorithm function independently of the receiver's decisions, employing at all times the actual transmitted data symbol values. In normal transmission, however, it is intended that the tracking algorithm employ the receiver's decisions. The investigations to be reported in this chapter assume this to be the case, unless the receiver is being "trained", in which case the transmitted symbols are already known.

The results of the previous chapter showed us that around $P_e \lesssim 10^{-3}$ the MLSE/near-MLSE is only slightly better than the DFE-II$_2$, when both devices use the same channel estimate. Assuming adequate provision is made in the channel estimation process to compensate for the delay in detection of the MLSE/near-MLSE, it is reasonable to assume that its behaviour in mode III will not be too dissimilar to that of the DFE. We shall therefore confine our attention to the DFE only in this chapter. An interesting result of the previous chapter is that the MLSE is only as good as, and the near-MLSE is inferior to, the DFE-II$_1$. In chapter 5 it was shown that around $P_e \lesssim 10^{-3}$ the possible advantage of any device over the DFE-I$_1$ is only ~1–3 dB (in terms of *SNR*) anyway. It therefore seems highly desirable, and more cost-effective than employing a near-MLSE, to employ a DFE in which serious error propagation is prevented by some arrangement, such as use of a training sequence from the transmitter. In any case, as we shall shortly see, the effect of bursts of errors such as occur for 16 and 64 QAM can prove very trying for the tracking algorithm, and so recourse to training from the transmitter is likely to be needed anyway.

In this chapter two approaches to use of training sequences from the transmitter will be examined; one is where a sequence of known symbols is inserted periodically into the data stream, the other where a training sequence is requested by the receiver.

In all measurements the fade rate is set at 1.0 Hz, and the warm-up time for the tracking algorithm is 1.67 fade periods, or 4000 known symbols, prior to transmission over the fading patterns.

8.1 Performance of DFE(6,g)-III

Recall from section 3.3.3 what was said about a receiver operating in decision directed mode. The experience of deep fades at the receiver, when the instantaneous error rate is high, or where the channel undergoes rapid changes in time, can sometimes perturb the detection and tracking process in such a way that, on "recovery" out of the difficult period, the detected data symbols are shifted by a phase angle $i\pi/2$, where i=1, 2 or 3, with respect to the transmitted symbols, e.g. see [15] or [64]. The estimate of the channel vector likewise has all its components shifted by a phase angle $-i\pi/2$ with respect to the true channel vector. This state of affairs causes no problem to the reception of data if the information on which quadrant (in the complex plane) the k^{th} data symbol is in (that is, the symbol that is intended for the customer as opposed to the one that is actually transmitted) is encoded as the quadrant change between the s_{k-1} and s_k transmitted data symbols. For example, the receiver passes on to the customer the k^{th} data symbol given as

$$\text{Customer's data symbol} = \hat{s}_k e^{-j\mathrm{Qu}(\hat{s}_{k-1})\pi/2} \tag{8.1}$$

where Qu(.) is an integer function taking on the values

$$\left.\begin{aligned} \mathrm{Qu}(\hat{s}_k) &= 0 \quad \text{for} \quad 0 < \arg(\hat{s}_k) < \tfrac{\pi}{2} \\ &= 1 \quad \text{for} \quad \tfrac{\pi}{2} < \arg(\hat{s}_k) < \pi \\ &= 2 \quad \text{for} \quad \pi < \arg(\hat{s}_k) < \tfrac{3\pi}{2} \\ &= 3 \quad \text{for} \quad \tfrac{3\pi}{2} < \arg(\hat{s}_k) < 2\pi \end{aligned}\right] \tag{8.2}$$

It can be seen from (8.1) that any fixed quadrangular phase shift between the detected and transmitted symbols does not alter the symbol passed on to the customer. The strategy of coding data as the phase difference between two adjacent (in time) symbols is termed differential coding. We shall in future distinguish the symbols and symbol errors detected by way of differential coding with the descriptor "differential", as opposed to the ones arising out of actual threshold detection, which have hitherto been referred to simply as "detected". Given the scheme of (8.1), it can be seen that the number of differential symbol errors is $\geq$ (only $>$ for 4 QAM) the number of detected symbol errors, by at most a factor of 2, assuming there is no phase shift between the detected and transmitted symbols. The difference is usually greater for small constellation sizes, since an erroneous detected symbol that is a "nearest neighbour" to the transmitted symbol is more likely to be in a different quadrant, this always being the case for 4 QAM.

Figs. 8.1 and 8.2 depict the P_e(E.C.) vs. *SNR* performance of the DFE(6,*g*)-III, with the error rates being derived from the differential error counts that occurred from use of the scheme in (8.1). Also shown in figs. 8.1 and 8.2 are the curves of P_e(E.c.) vs. *SNR* based on the differential error counts of the DFE-II$_2$. Tables 8.1 and 8.2 list the differential error counts of the DFE, using the SRK algorithm, for both mode II$_2$ and mode III, and also lists the actual detected error counts too. Comparing the counts for the detected errors and the differential errors, where the former is significantly greater than the latter tells us that a quadrangular phase shift in the detected symbols, with respect to the transmitted symbols, occurred at some time during the measurement run.

The performance of the DFE-III using the SD algorithm is not radically different from that with the SRK algorithm. The distinct "random" variations in the curves for each algorithm may be different, as shown in figs. 8.1 and 8.2, but essentially they both portray the same behaviour. For the rest of this chapter the discussion is centred on the SRK algorithm. Fig. 8.1 shows that the degradation of mode III with respect to mode II$_2$ is very slight with 4 QAM on Channel A. The difference between the two modes steadily narrows as the *SNR* increases/error rate decreases. With 4 QAM on Channel B (fig. 8.2), the degradation of mode III with respect to mode II$_2$ is larger than for Channel A, particularly above $P_e = 10^{-3}$, but this narrows sharply as the *SNR* increases. At $P_e = 10^{-3}$ for 4 QAM, the degradation of mode III with respect to mode II$_2$ is $\sim$0.5 dB in terms of *SNR*, for both channels.

SNR (dB)	No. of errors in 60,000 for DFE(6,1)			
	Mode II_2		Mode III	
	Threshold	Differential	Threshold	Differential
4 QAM	Detected	Detected	Detected	Detected
7.5	5,986	8,773	43,068	11,250
10.0	3,028	4,357	42,477	6,474
12.5	1,316	1,902	29,659	3,708
15.0	603	859	59,587	973
20.0	154	214	17,384	230
24.0	29	47	17,304	62
27.5	6	10	6	10
30.0	3	4	3	4
16 QAM				
15.0	11,276	12,652	53,475	38,829
20.0	2,993	3,267	55,995	22,811
25.0	615	667	59,587	4,810
30.0	175	195	17,344	1,687
32.5	77	86	17,264	1,611
35.0	16	19	48	52
37.5	13	15	1,397	1,449
40.0	6	7	6	7
64 QAM				
25.0	7,964	8,187	58,231	49,677
30.0	1,804	1,850	56,254	38,829
35.0	472	485	59,724	14,560
37.5	284	288	19,149	5,274
40.0	176	177	37,597	6,779
42.5	128	128	1,787	1,795
45.0	100	101	17,225	9,229

Table 8.1 Error counts for DFE(6,1) on Channel A, using SRK algorithm.

SNR (dB)	No. of errors in 60,000 for DFE(6,2)			
	Mode II_2		Mode III	
	Threshold	Differential	Threshold	Differential
4 QAM	Detected	Detected	Detected	Detected
7.5	7,164	10,523	52,863	18,930
12.5	1,497	2,129	54,094	5,083
16.5	274	372	16,871	1,449
20.0	35	55	41	68
22.5	12	21	12	21
16 QAM				
17.5	9,422	10,596	56,774	44,376
22.5	2,061	2,271	42,477	29,319
26.5	369	406	59,450	7,837
30.0	57	58	1,549	1,622
31.0	32	32	16,794	2,356
64 QAM				
25.0	16,602	17,146	59,177	53,598
30.0	6,444	6,594	58,950	57,609
32.5	1,863	1,893	44,787	31,634
35.0	505	518	58,905	57,564
37.5	103	108	58,769	56,124
40.0	22	22	17,028	9,642

Table 8.2 Error counts for DFE(6,2) on Channel B, using SRK algorithm.

For 16 QAM, the gap between mode II_2 and mode III is significantly larger than that for 4 QAM, since the effect of error propagation is generally more severe. The mode III curves do show an inclination to converge towards the mode II_2 curves, but in quite an erratic fashion. Note, for example, the sudden rise in error rate at *SNR*=37.5 dB on Channel A, and *SNR*=31.0 dB on Channel B. Measurements were conducted at these two *SNR*'s with different values of ω and noise sample sequences. The results are shown in Table 8.3. Observe in Table 8.3 the marked difference that occurs for 16 QAM in mode III at *SNR*=37.5 dB on Channel A, when the tracking parameter ω is increased slightly, or when the noise sequence is changed. At *SNR*=31.0 dB on Channel B, changing the noise sequence has had little effect in mode III on the number of differential errors that occur, while a slight increase in ω has brought about a substantial reduction. One might expect that a change in the random noise sequence or the random variation of the channel estimate (caused by a slight change in ω) tends to be more influential as the error count for the DFE-II_1 gets lower. The figures in Table 8.3 show that the error count for the DFE-III, in contrast to that for the DFE-II, is much more sensitive to any short-term random variations. One can imagine that at a particularly sensitive time instant, and assuming the detection and tracking process is unaffected by decision errors up to that point, the occurrence of just one error in the DFE-III can lead to a large enough burst of errors that perturbs the detection and tracking process to an alarming degree.

In general, having a lower value of ω in the SRK algorithm means that the tracking process is more susceptible to short-term decision errors, because the window size is smaller. Thus when a low value of ω is required to achieve the necessary tracking accuracy; as it would be for a large constellation size, the effect of decision errors is also more likely to cause error in the channel estimate. This problem is then greatly compounded for large constellation sizes by the fact that propagative error bursts tend to be much more severe. For 64 QAM the effect of error bursts is, as expected, more severe than it is for 16 QAM, as can be seen from figs. 8.1 and 8.2; the error rate can barely be reduced below 10^{-1}. Increasing ω, or having a different noise sequence, does not appear to relieve matters much, as Table 8.3 shows.

	No. of errors in 60,000 for DFE(6,g)			
ω	Mode II$_1$	Mode II$_2$	Mode III	
16 QAM, Ch. A SNR=37.5 dB	Threshold Detected	Differential Detected	Threshold Detected	Differential Detected
0.754 (opt.)	7	15	1,397	1,449
0.754 †	1	7	16	16
0.794	7	15	14	16
16 QAM, Ch. B, SNR=31.0 dB	(† indicates different noise sequence)			
0.864 (opt.)	5	32	16,794	2,356
0.864 †	5	46	2,467	2,378
0.874	5	32	99	105
64 QAM, Ch. A, SNR=45.0 dB				
0.591 (opt.)	11	101	17,225	9,229
0.591 †	7	77	17,344	2,859
0.791	10	84	2,925	2,912
0.791 †	9	82	2,859	2,832
0.950	455	1,767	58,097	34,368
64 QAM, Ch. B, SNR=40.0 dB				
0.741 (opt.)	2	22	17,028	9,642
0.741 †	3	86	59,862	3,237
0.841	4	63	17,146	2,787
0.841 †	9	101	59,724	8,936
0.950	336	5,002	58,499	51,541

Table 8.3 Error counts for 16 and 64 QAM with variations in ω and noise sequences.

It would appear that with 16 QAM, performance in mode III with $P_e \lesssim 10^{-3}$ is only possible when the errors from the DFE-II$_1$ are very small to begin with; with 64 QAM they probably have to be zero. Referring back to chapter 7, the curves in figs. 7.11 and 7.13 inform us that for the DFE-II$_1$ on Channel A, an *SNR* of about 50.0 dB or more is needed to produce a P_e of 10^{-6} for 64 QAM at a 1.0 Hz fade rate, while on Channel B the corresponding value is about 45.0 dB. For a 2.0 Hz fade rate the required *SNR*'s will naturally be greater. Clearly then, relatively large transmitter powers are needed if the DFE-II$_1$ is to produce very low error rates with 64 QAM. Even then, one error may be enough to cause a significantly harmful error burst. Lower fade rates than 1.0 Hz would perhaps yield more acceptable performance with 64 QAM, because then ω can be set to a high value, say $\gtrsim 0.95$. However, for $\omega = 0.95$ to be optimum, in the sense of minimizing the error in the channel estimate, a fade rate of about 0.03 Hz is required at $SNR = 45.0$ dB on Channel A, and a fade rate of about 0.08 Hz is required at $SNR = 40.0$ dB on Channel B; thus it can be seen that the fade rate would have to be substantially lower than 1.0 Hz.

The effect of error bursts on the tracking algorithm is so severe with 64 QAM that, on many occassions, the measurement runs for this constellation size finished with the receiver in a "nonaligned" state, i.e. the tracking algorithm was widely "off the mark" and the detector was producing errors. The fading patterns used in the measurement runs (figs. 4.5 and 4.6 of chapter 4) ensure that the instantaneous signal-to-noise ratio SNR_k (4.18) finishes on a high value. The measurement runs for 4 and 16 QAM always managed to finish with the receiver in an "aligned" state.

Overall, the results for the DFE-III suggest that if the receiver is to operate at fade rates as high as 1.0 Hz, without any arrangement by which the tracking can re-align itself, then 16 QAM may perhaps be the largest rectangular constellation size that can feasibly be used, providing a data rate of 9.6 kbits s^{-1}, and only then with an *SNR* that gives a sufficiently low error rate for the DFE-II$_1$.

One might think that the closeness of the mode II_2 and mode III curves for 4 QAM is due to the relatively short propagation bursts that occur with this constellation size, ensuring that the tracking algorithm is not perturbed by decision errors for too long. This may be only one reason. Another reason may come from the highly desirable property of phase constellations, as 4 QAM is, of enabling "self-recovery" of a tracking algorithm. What this means is that the receiver is able to self-adapt to channel conditions without needing to know what the transmitted symbols are, i.e. no training is required. The reality of self-recovery has been demonstrated for the start-up convergence of adaptive linear equalizers on time-invariant channels [83], [89]–[91], using the SD algorithm, and mathematical analyses have been put forward to explain it [90], [91]. For the fading channels used in this thesis, it has been observed with 4 QAM that for both the DFE (via the CE method) and MLSE, using either the SD or SRK algorithm, the receiver is able to self-align from start-up in at most ~100 symbol intervals (with a possible quadrangular phase shift). In general, the time taken to self-align is dependent on the initial value of the estimated vector, the number of levels in the data signal, and the amount of ISI introduced by the channel. Unlike for adaptive linear equalizers, it seems difficult to prove that self-recovery can take place for an adaptive DFE or MLSE, and more so if the SRK algorithm is used, because of the more complex inter-dependency between adjacent (in time) symbol decisions. Nevertheless, assuming that it can occur, transmission with 4 QAM is thus desirable because it does not need to rely on training sequences, unlike 64 QAM does, in order to recover out of a difficult period like a deep fade. Self-recovery perhaps also plays a limited role in transmission with 16 QAM, since this constellation can be crudely thought of as a phase constellation 75% of the time (consider outer points only). It is possible to devise a self-recovery scheme for combined PAM-PSK type constellations, as is detailed for example in [90]–[92], by approximating the non-phase constellation by a phase constellation. This has been successfully demonstrated for linear equalization in [90], [91] on time-invariant channels, using an SD-type algorithm. However, approximating a non-phase constellation by a phase constellation naturally introduces a certain amount of self-noise, this being larger the greater the degree of PAM is present. The effect of this self-noise would be compounded in a

DFE through the feedback taps. It is not evident whether a self-recovery scheme on HF channels could be devised for 64 QAM using a DFE, remembering that there is also the question of convergence time. The convergence times shown in [90], [91] for 16-point and 32-point constellations are in the region of 10^3–10^4 symbol intervals, which is probably too long for the time-varying channels looked at here. For further information on the interesting area of self-recovery, see [89]–[92].

To round-off this section we will mention what happened when the DFE-III was implemented by way of the conventional method, using the SRK algorithm and with the same *SNR* values as was used for the CE method of implementation. Not surprisingly, performance was fouund to be poorer, but in a more dramatic way than expected. All the measurement runs, for 4, 16 and 64 QAM on Channel A and Channel B, produced a $P_e(\text{E.C.}) > 10^{-1}$, the one exception being at SNR=22.5 dB on Channel B with 4 QAM, which produced 27 differential errors and 19 detected errors. At the end of all the runs, bar the one exception, the receiver was in a non-aligned state, and in a majority of these it was noticed that the tracking algorithm was in a state of instability, i.e. one or more elements of the $\mathbf{D}_k$ matrix continuing to increase at each update. In fact, one of the measurement runs did not complete the full 60,000 symbol transmission, because the computer's upper limit for a real number had been exceeded, thus causing the program to abort. We ascertained that the cause of instability was from the estimated vector reaching a constant setting, in which the feedforward taps of the DFE were all zero, and the feedback taps were such that the detected symbols were cyclically repeating themselves in a manner which ensured the algorithm error $\epsilon_k = 0$ always. For example, the measurement run which aborted was with 16 QAM at SNR=26.5 dB on Channel B. The value of the last element of the diagonal matrix $\mathbf{D}_k$, i.e. d_8, had exceeded the upper limit on the computer, all the other elements being between 0.1 and 0.5. The estimated vector at the time of abortion was (with the feedback tap values accurate to 3 d.p.),

$$\underset{\sim}{c}_k^t = [0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ -(0.527+j4.304)\ ,\ -(5.304-j0.527)] \tag{8.3}$$

and the detected symbols, which repeated themselves every 4 baud intervals, were in the sequence $(1-j)$, $(-1-j)$, $(-1+j)$, $(1+j)$, going from $\hat{s}_k$ to $\hat{s}_{k-3}$. It can be seen that $\epsilon_k=\hat{s}_k-\underline{x}_k^t\underline{c}_{k-1}=0$ always, so that the estimated vector remains fixed at the value in (8.3). Since the feedforward taps are all zero, the received samples $\{r_i\}$ are not used, thus permitting $\underline{c}_k$ to take on a constant value. The estimated vector can be viewed as having reached a steady-state solution to the least squares criterion of (3.33), given that the symbol values used in the input vector $\underline{x}_k$ are not restricted to being the actual transmitted symbols, but free to take on any values they like. It is easy to see that given the sequence of detected values $(1-j)$, $(-1-j)$, $(-1+j)$, $(1+j)$, the RLS solution for $\underline{c}_k$ that satisfies $\epsilon_k=0$, with the feedforward taps all zero, is not unique, and the $\mathbf{P}_k^{-1}$ matrix under such circumstances is singular; this therefore makes the determinant of $\mathbf{D}_k$ infinite.

When implemented by way of the conventional method, the DFE-III may go through a long period of say 10,000 symbols or more, before it encounters a sufficiently abrupt change or deep fade that causes the whole detection and tracking process to assume something like the undesirable state we have just described. The possibility of this situation arising does not appear to have been mentioned much in the literature, maybe because the time spans observed are too short. To stop the situation arising, it would probably be necessary to have a constraint preventing some or all of the feedforward taps from settling to a zero value. We note that a similar state of affairs cannot arise in channel estimation; the estimated vector could never settle to a fixed value, with $\epsilon_k=0$, because the "random" received samples are never prevented from influencing the proceedings.

8.2 DFE-III with Periodic Training Sequence (PTS)

Consider a scheme in which a finite sequence of symbols, known to the receiver, is inserted periodically into the transmitted sequence of data symbols. If the gap between the finish and start of two successive training sequences is x symbols, and if the length of a training sequence is y symbols, then the scheme is described as being a $100y/(x+y)\%$ PTS scheme. It is assumed that the receiver knows the start and finish of the training sequence.

In this section we shall observe the performance of the DFE-III when a 10% PTS scheme is employed ($x=90$, $y=10$). At the start of each training sequence the channel estimate vector is set to zero, and the $\mathbf{U}_k$ and $\mathbf{D}_k$ matrices in the SRK algorithm are set to the unit identity matrix. A sequence of 10 known symbols is an adequate time to allow for the DFE, implemented by way of the CE method with $\eta=10^{-2}\sigma_s^2$, to align itself, as was shown in chapter 7. The training sequence we use for Channel A is made up of a 2-symbol repeating sequence, while for Channel B it is made up of a 3-symbol repeating sequence, these given respectively as

$$m\{\ (1+j)\ ,\ (-1+j)\ \} \tag{8.4}$$

$$m\{\ (1+j)\ ,\ (-1+j)\ ,\ (1-j)\ \} \tag{8.5}$$

where we set $m=1$, 3 and 5 for 4, 16 and 64 QAM respectively to ensure a signal power $\geq\sigma_s^2$. The two sequences in (8.4) and (8.5) are desirable because each one is linearly independent of any cyclically-shifted version of itself. Thus, for both channels, the $g+1$ successive input vectors $\underline{\mathbf{x}}_k, \ldots, \underline{\mathbf{x}}_{k-g}$ to the SRK algorithm are linearly independent. The sequence in (8.4) also happens to be self-orthogonal, i.e. the scalar product of two different cyclically-shifted sequence vectors is zero, which makes it suitable for the start-up channel estimation process described in [84], although this is not utilized here.

Figs. 8.3 and 8.4 depict the P_e(E.C.) performance of the DFE-II$_2$ and DFE-III with a 10% PTS scheme. Table 8.4 lists the error counts that occurred during the measurement runs, which are only for the periods during which no training symbols were being received. The error counts are based on straightforward threshold detection of the symbols, because these were found to be always lower than the differentially detected errors. This indicates that the receiver never had time, between training sequences, to become aligned with a quadrangular phase shift of $i\pi/2$. The curves for 4 QAM show that the performance of the DFE-III is very close to that of the DFE-II$_2$. On Channel A with 16 QAM, the curves for the DFE-III and

SNR (dB) (Channel A)	No. of errors in 60,000 for DFE(6,1)	
4 QAM	Mode II$_2$	Mode III
7.5	5,986	6,182
10.0	3,028	3,000
12.5	1,316	1,439
15.0	603	631
20.0	154	156
24.0	29	29
27.5	6	7
30.0	3	1
16 QAM		
15.0	11,276	11,672
20.0	2,993	3,297
25.0	615	589
30.0	175	191
32.5	77	67
35.0	16	19
37.5	13	8
40.0	6	1
64 QAM		
25.0	7,964	10,284
30.0	1,804	2,389
35.0	472	550
37.5	284	278
40.0	176	240
42.5	128	187
45.0	100	95

SNR (dB) (Channel B)	No. of errors in 60,000 for DFE(6,2)	
4 QAM	Mode II$_2$	Mode III
7.5	7,164	7,659
12.5	1,497	1,664
16.5	274	252
20.0	35	35
22.5	12	10
16 QAM		
17.5	9,422	11,224
22.5	2,061	3,077
26.5	369	460
30.0	57	55
31.0	32	24
64 QAM		
25.0	16,602	20,191
30.0	6,444	7,819
32.5	1,863	3,252
35.0	505	773
37.5	103	292
40.0	22	52

Table 8.4 Error counts for DFE(6,g)-II$_2$, and DFE(6,g)-III with 10% PTS scheme.

DFE-II$_2$ are quite close together in the high P_e region, but then start to differ, though not drastically so, around SNR=32.5 dB to 37.5 dB. As we saw in the last section, the error rate of the DFE-III in this region is highly sensitive to the short-term random variations of the channel estimate and noise samples. With 64 QAM on Channel A, the DFE-III curve looks like a slightly offset version of the DFE-II$_2$ curve. On Channel B the difference between the DFE-III and DFE-II$_2$ is reasonably small for both 16 and 64 QAM.

On the whole, then, we can say that a PTS scheme is effective in allowing performance close to, or occassionally even better than, the DFE-II$_2$ to be achieved. The difference between the DFE-III-with-PTS and DFE-II$_2$ tends to be larger for the larger constellation size.

It was found that the periodic re-setting of the channel estimate vector and the $\mathbf{U}_k$ and $\mathbf{D}_k$ matrices of the SRK algorithm did not make much difference from the situation whereby they are just left alone. If a Fast Kalman algorithm is used, and/or if limited precision processing is employed, then re-setting is particularly desirable to avoid the cumulation of numerical round-off errors.

8.3 DFE-III with Request for Training Sequence (RTS)

Consider a scheme whereby the receiver, deciding that data is not being detected reliably enough, initiates the transmission of a training sequence from the transmitter. When the receiver subsequently decides that data is being detected reliably enough, another signal is sent to the transmitter requesting the resumption of normal data transmission. In theory, therefore, the likelihood of serious error propagation can be reduced, and performance close to that of the DFE-II$_1$ achieved. This scheme assumes that a feedback link exists from the receiver to the transmitter, though it does not need to be very sophisticated, as it only needs to carry a simple ON/OFF type signal.

In order to decide what the current level of performance is, the receiver needs to monitor some performance index. An obvious one that comes to mind is the error at the output of the equalizer [64], [71]. Consider the quantity

$$H_k = (1-\lambda)H_{k-1}+\lambda|\tilde{s}_k-\hat{s}_k|^2 \tag{8.6}$$

where λ is in the range $0<\lambda<1$, and governs how far back into the past the "averaging" of H_k extends (window size $\sim 1/\lambda$). The quantity $\tilde{s}_k$ is the pre-threshold estimate of s_k. Whenever H_k, computed at the k^{th} time instant, is above a threshold level H_{t}, the receiver sends a signal to the transmitter and thereby initiates the transmission of a training sequence. The training sequence sent by the transmitter, for Channel A and Channel B respectively, is the 2-symbol and 3-symbol repeating sequence of (8.4) and (8.5). While training symbols are being received, the receiver continues to compute H_k, and when $H_k<H_{\mathrm{t}}$ a signal is sent to the transmitter requesting that normal data transmission be resumed. It is assumed that the receiver knows the start of the training sequence, although this is not absolutely necessary for convergence to be achieved [84].

Figs. 8.5 and 8.6 depict the performance of the DFE-II$_1$ and DFE-III with an RTS scheme, the P_e being measured by means of the analytical bound mentioned in chapter 5. The threshold H_{t} is set at values of 0.1 and 0.2, and λ is set at values of 0.03 and 0.09. The figures next to each plot-point in figs. 8.5 and 8.6 refer to the percentage of the 60,000 baud time for which training sequences were transmitted. After having produced the detected symbol $\hat{s}_k$, if $H_k \geq H_{\mathrm{t}}$ $(H_{k-1}<H_{\mathrm{t}})$ then the r_{k+N} received sample contains the first training symbol; if, on the other hand, $H_k<H_{\mathrm{t}}$ $(H_{k-1}\geq H_{\mathrm{t}})$ then r_{k+N} contains the first non-training symbol. After H_k is first monitored to be $\geq H_{\mathrm{t}}$ $(H_{k-1}<H_{\mathrm{t}})$, we stop monitoring H_k (but continue to evaluate it) until the first training symbol arrives for "detection", and then when H_k is found to be $<H_{\mathrm{t}}$ we stop monitoring H_k (but continue to evaluate it) until the first non-training symbol arrives. We shall define the "training period" as that period from when $H_k \geq H_{\mathrm{t}}$ $(H_{k-1}<H_{\mathrm{t}})$ to when the first non-training symbol arrives; the P_e measure excludes the training periods. The training period of course neglects the time taken for the feedback signal to reach the transmitter, and the delay of the signal from transmitter to receiver. It was found that very few detected errors

from the DFE-III-with-RTS occur during the non-training periods, and consequently this is why the analytical bound is employed, proving to be very useful. The bound assumes, of course, that the feedback symbols in the DFE are correct. Detected errors occurred only with $H_{\mathrm{t}}=0.2$, and the counts are listed in Table 8.5 along with those for the DFE-II$_2$, for ease of comparison.

The curves of figs. 8.5 and 8.6 show that an RTS scheme offers substantial improvements in performance. For example, in fig. 8.5 we see that with 4 QAM at $SNR=20.0$ dB on Channel A, a P_e of 10^{-6} can be achieved at a power saving of about 13 dB (with respect to the DFE-II$_1$) with 10% training; with 16 QAM at $SNR=30.0$ dB, a P_e below 10^{-6} can be achieved at a power saving of about 13 dB with approximately 5% training; with 64 QAM at $SNR=40.0$ dB, a P_e of nearly 10^{-6} is possible with only 2.5% training. On Channel B (fig. 8.6) the power saving is generally lower than that for Channel A, but still significant. For example, with 4 QAM at $SNR=22.5$ dB on Channel B, a P_e of about 3×10^{-7} can be achieved at a power saving of 5 dB with 4.4% training; with 16 QAM at $SNR=30.0$ dB, a P_e below 10^{-6} can be achieved at a power saving of 6 dB with 7.6% training; with 64 QAM at $SNR=37.5$ dB, a P_e of approximately 2×10^{-6} is possible at a power saving of 6 dB with 8.6% training. Note that all the above quoted figures are for $H_{\mathrm{t}}=0.1$ and $\lambda=0.03$. It appears from figs. 8.5 and 8.6 that it is better, in general, to have λ small rather than large, as can be seen by comparing the two curves for $\lambda=0.03$ and 0.09 with $H_{\mathrm{t}}=0.1$.

The performance with RTS on Channel A often yields lower error rates with lower training times, at a given SNR, compared to those for Channel B. This may not be a totally fair comparison to make, however, as it should be remembered (from chapter 5) that for a ZF MFE, a longer observation time than 25 fade periods can increase the SNR required to achieve a P_e below 10^{-5} (recall figs. 5.1 and 5.3), this being more so for Channel A than Channel B. Another point to consider, though, is that the RTS scheme would be expected to eliminate the deeper fade periods from the useful transmission time. Thus, given that the total average energy is the same for both channels, the comparison between them when an RTS scheme is

SNR (dB) (Channel A)	No. of errors in 60,000 for DFE(6,1)	
4 QAM	Mode II_2	Mode III
10.0	3,028	18
15.0	603	6
20.0	154	0
24.0	29	1
16 QAM		
20.0	2,993	19
25.0	615	14
30.0	175	1
35.0	16	0
64 QAM		
30.0	1,804	6
35.0	472	3
40.0	176	0
45.0	100	2

SNR (dB) (Channel B)	No. of errors in 60,000 for DFE(6,2)	
4 QAM	Mode II_2	Mode III
12.5	1,497	24
16.5	274	10
20.0	35	1
22.5	12	0
16 QAM		
17.5	9,422	4
22.5	2,061	10
26.5	369	17
30.0	57	3
64 QAM		
25.0	16,602	6
30.0	6,444	12
35.0	505	3
37.5	103	6

Table 8.5 Error counts for DFE(6,g)-II_2, and DFE(6,g)-III with RTS scheme (H_t=0.2, λ=0.03).

used is to a greater degree on the basis of the less dispersive (i.e. less ISI) Channel A against the more dispersive Channel B, and to a lesser degree on the basis of the extra diversity path of Channel B.

The use of an RTS scheme appears particularly attractive for 16 and 64 QAM, in light of the performances observed previously. Performance over an HF channel is governed largely by the deep fades that occur, relatively large transmitter powers therefore being needed to provide acceptable error rates. An important difference between HF channels and "bad" time-invariant channels, however, is that the former is not "bad" all of the time. The results in this section demonstrate that the time-varying channel is bad for only a modest proportion of the total transmission time, enabling substantial reductions in transmitter power when an RTS scheme is employed.

An important point to consider, depending on whether the particular application requires it, is the size of buffer needed by the transmitter for it to hold those data symbols whose transmission is delayed because the receiver is being trained. One way of enabling continuous throughput of data, thereby reducing the buffer size and increasing the overall useful information rate, is to transmit information using a smaller constellation size in place of the training symbols. For example, observe in fig. 8.5 that at SNR=35.0 dB the P_e for the DFE-II$_1$ with 4 QAM is less than 10^{-6}; thus with 16 QAM at SNR=35.0 dB, H_t=0.1 and λ=0.03, the training sequence could be reliably replaced with 4 QAM data signals, since there is little chance that these will be detected erroneously. Observe again, also in fig. 8.5, that with 64 QAM at SNR=45.0 dB, H_t=0.1 and λ=0.03, the training sequence could be reliably replaced with both 16 QAM and 4 QAM data signals. One could describe such a scheme as a *variable data rate* (VDR) scheme.

There are other designs for an RTS scheme that are possible. For example, another one, which is suitable for the MLSE too, is to monitor the squared magnitude of the error ϵ_k in the tracking algorithm, and also the squared magnitude of the received sample r_k, and thence produce a kind of signal-to-noise ratio estimate; when this drops below a threshold, a request-for-training signal is sent to the transmitter. Such a design was tested by us, and found to be not substantially different in performance from the one described earlier. One area for research is to investigate more efficient ways of forseeing performance deterioration, for it should be remembered that it only takes a few symbol errors arising in the DFE-II$_1$ for there to be very severe consequences, in the absence of training, for the DFE-III.

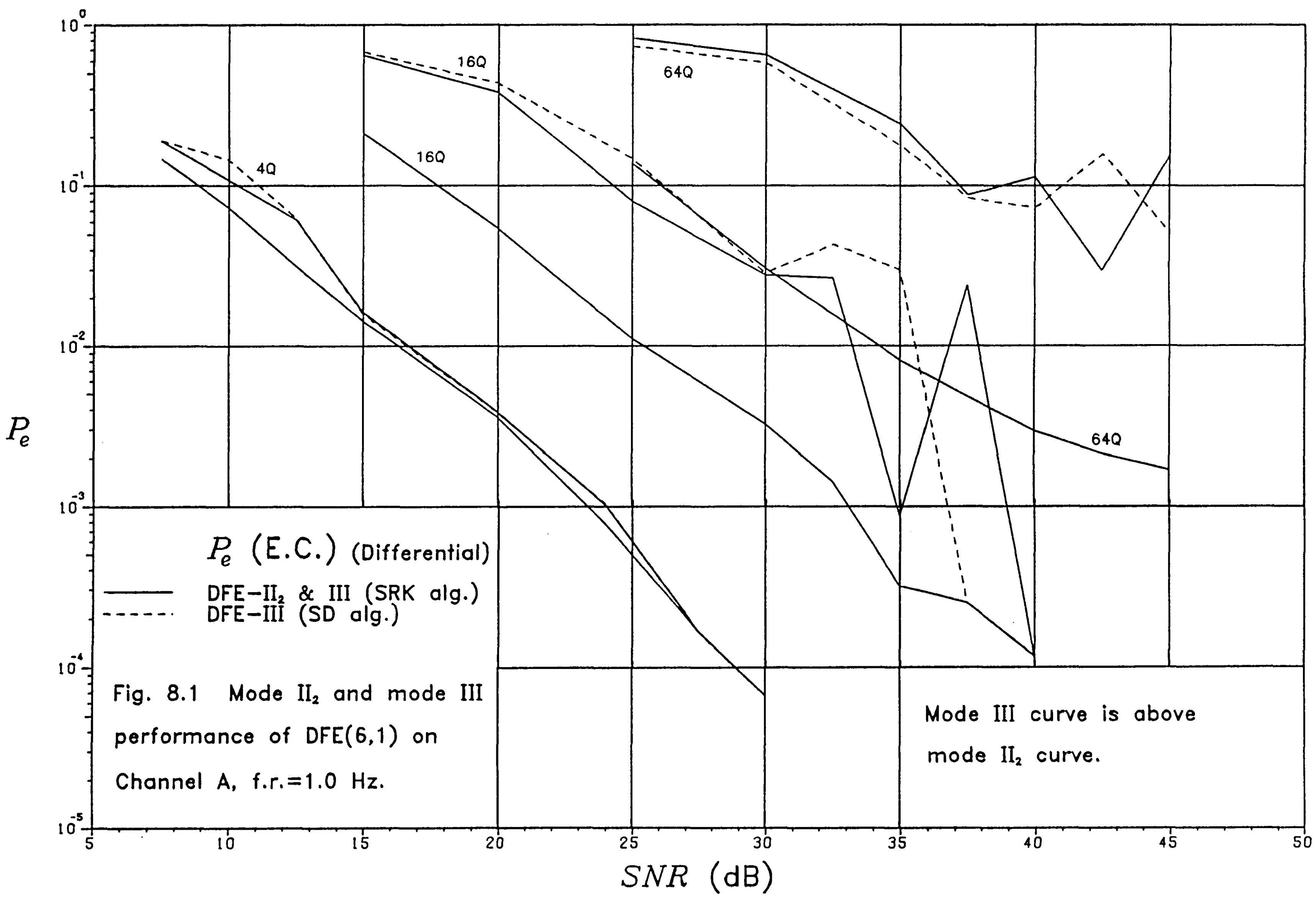


Fig. 8.1 Mode II_2 and mode III performance of DFE(6,1) on Channel A, f.r.=1.0 Hz.

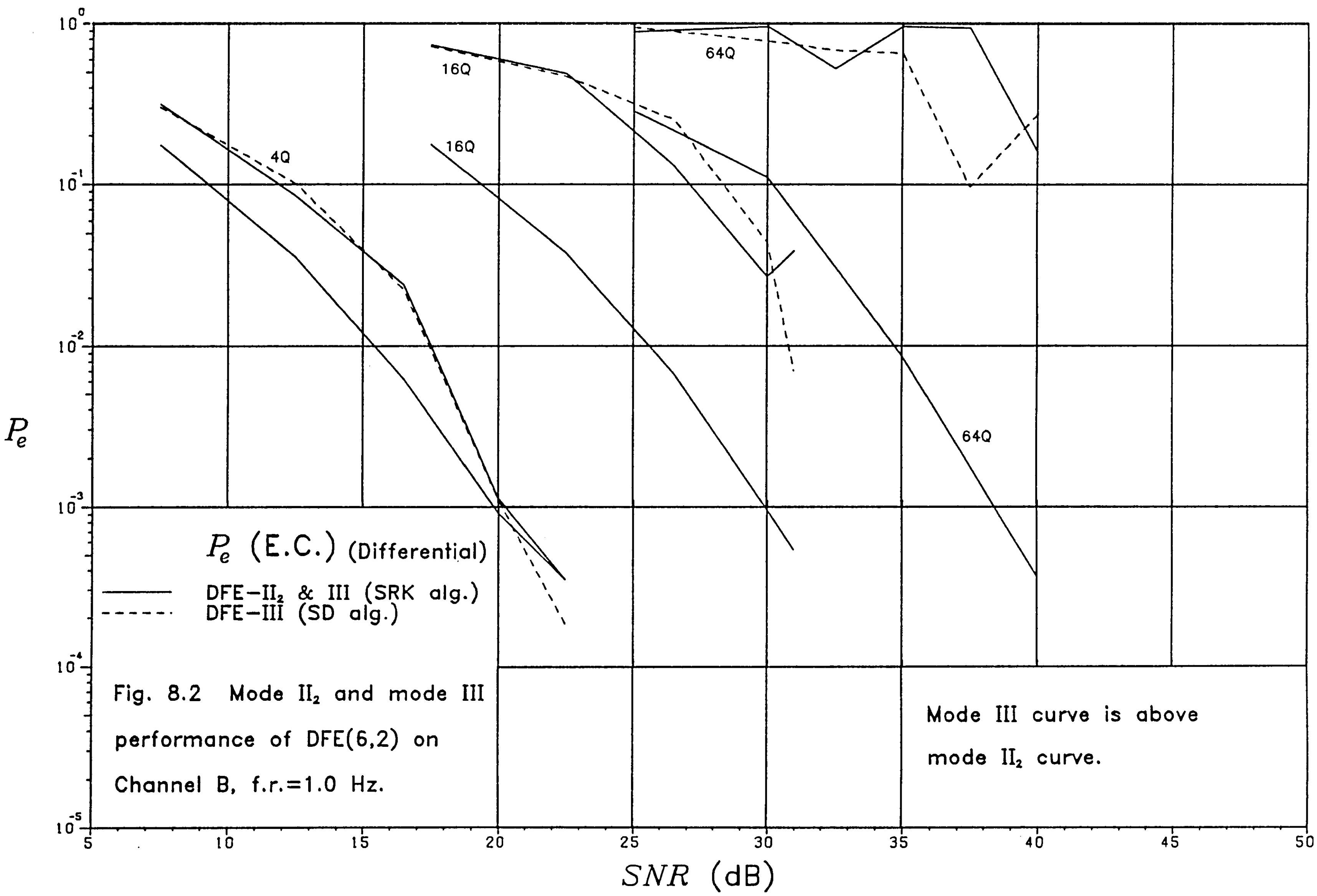


Fig. 8.2 Mode II_2 and mode III performance of DFE(6,2) on Channel B, f.r.=1.0 Hz.

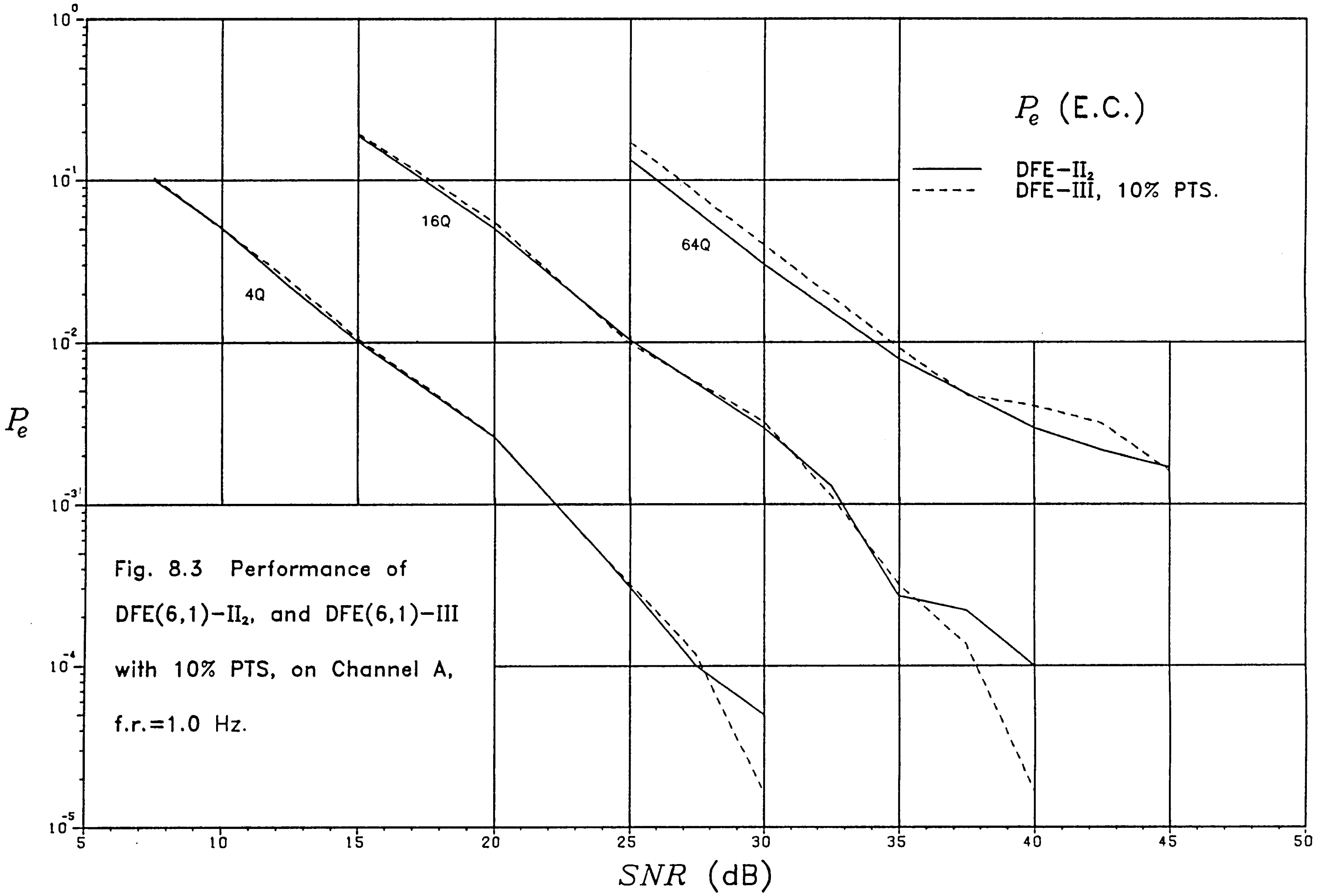


Fig. 8.3 Performance of DFE(6,1)−II_2, and DFE(6,1)−III with 10% PTS, on Channel A, f.r.=1.0 Hz.

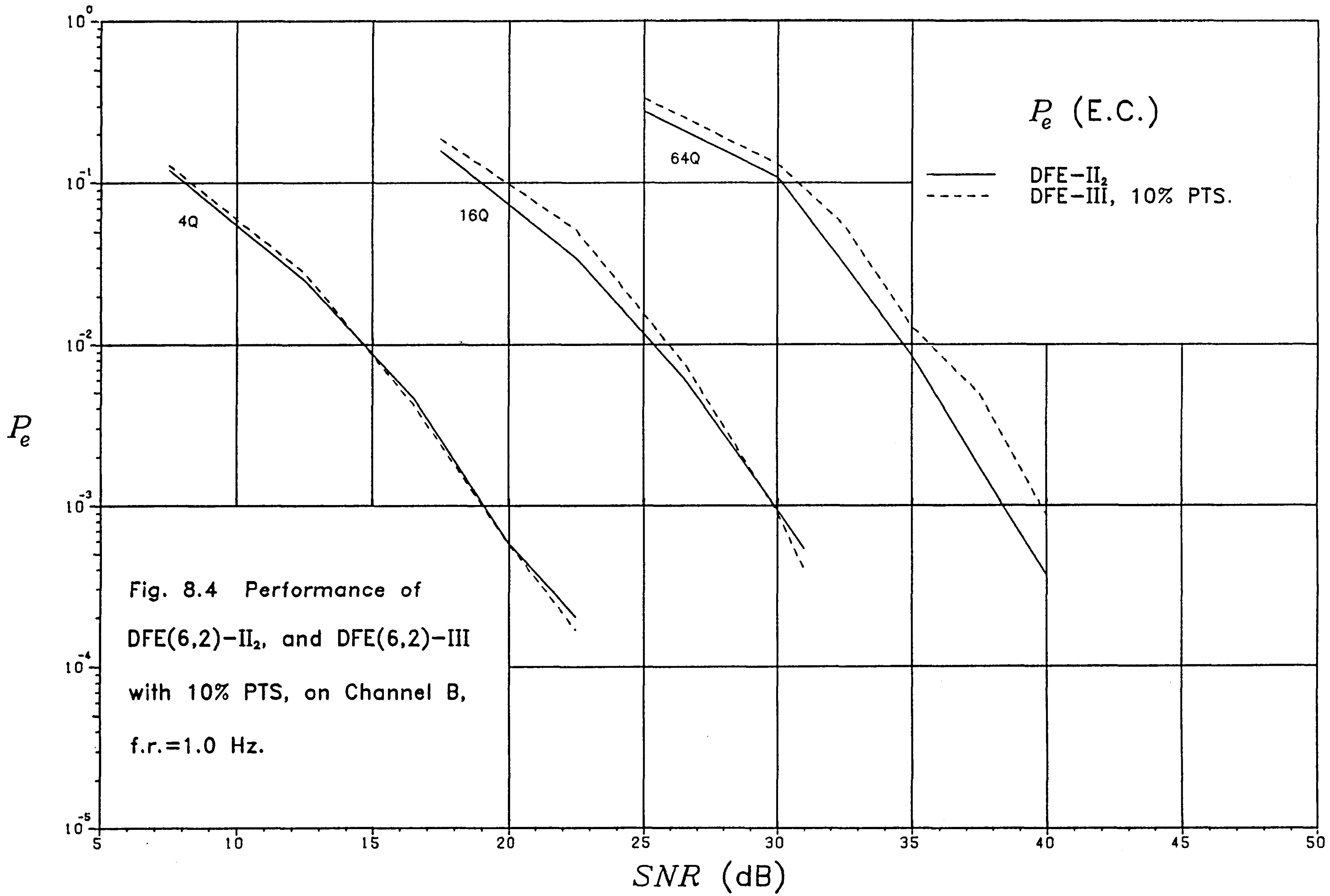


Fig. 8.4 Performance of DFE(6,2)–II_2, and DFE(6,2)–III with 10% PTS, on Channel B, f.r.=1.0 Hz.

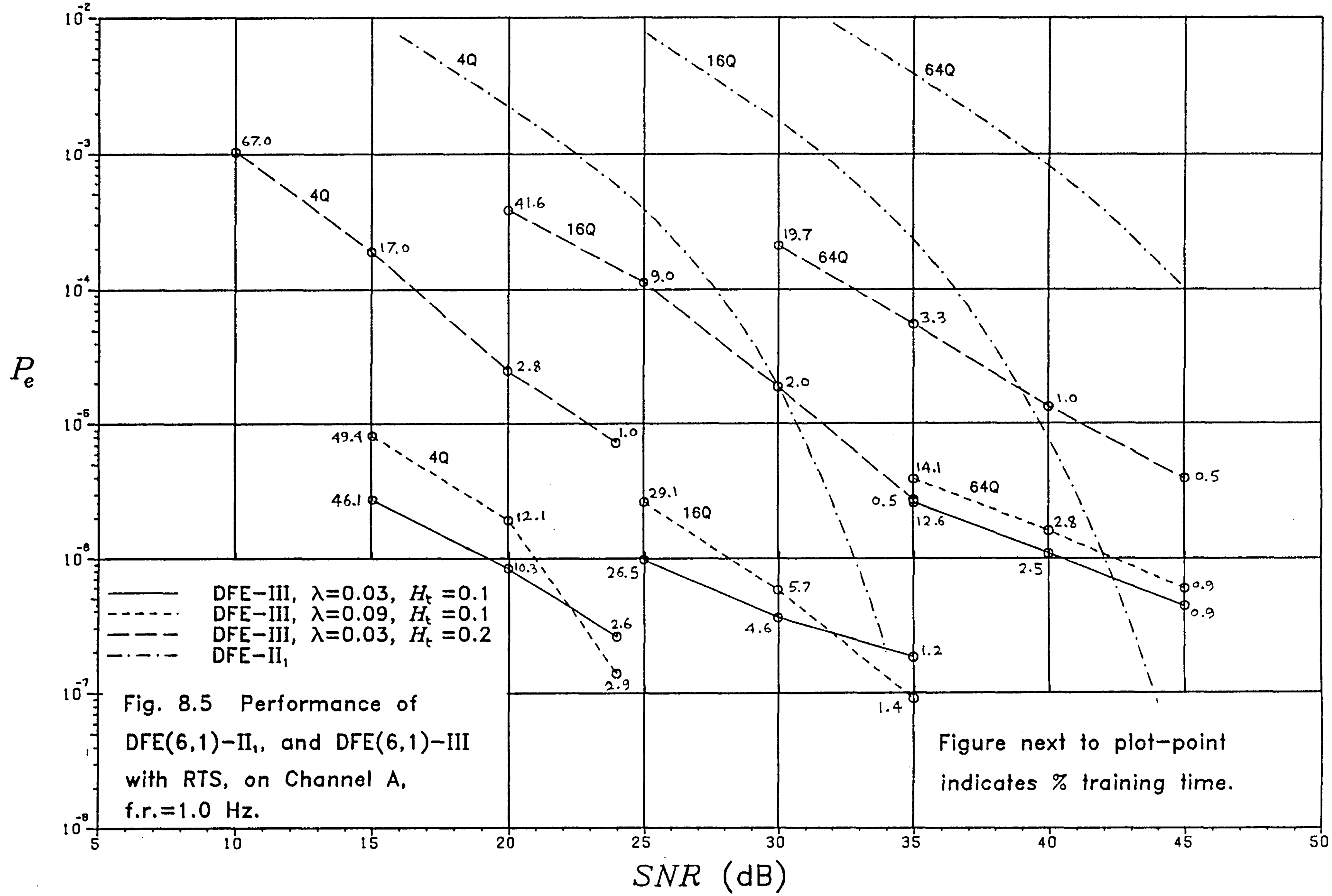


Fig. 8.5 Performance of DFE(6,1)–II$_1$, and DFE(6,1)–III with RTS, on Channel A, f.r.=1.0 Hz.

Figure next to plot–point indicates % training time.

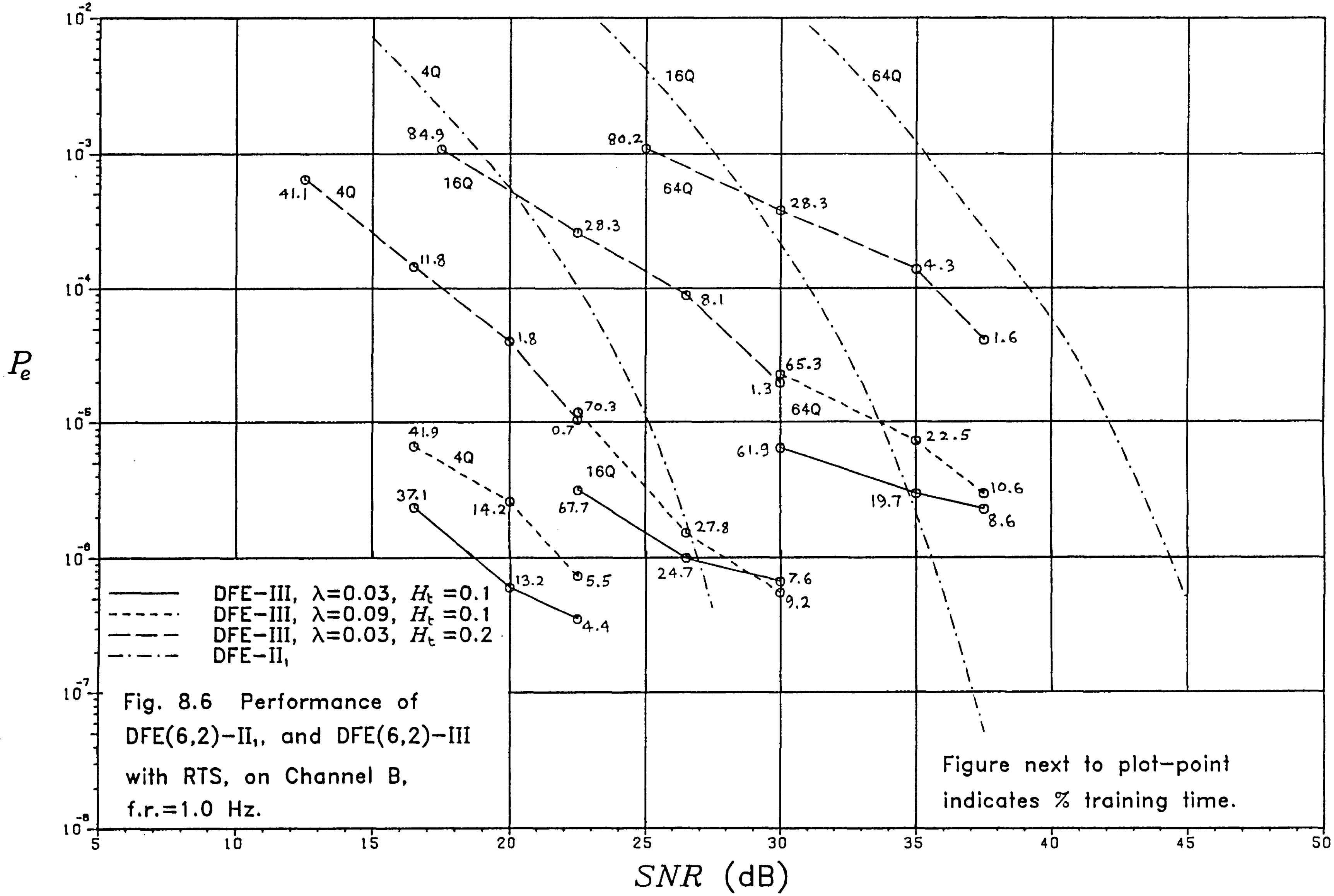


Fig. 8.6 Performance of DFE(6,2)-II$_1$, and DFE(6,2)-III with RTS, on Channel B, f.r.=1.0 Hz.

Figure next to plot-point indicates % training time.

Chapter 9
CONCLUSIONS

In this chapter we summarise the main points of the thesis, with some additional comments, and later give suggestions for further work.

In chapter 5 the performance of the receiver was observed for when adaptation is perfect. It was shown that the difference between the MMSE DFE and ZF MFE is ~1–3 dB in terms of *SNR*, this being achieved with a modest number of feedforward taps in the DFE. With a smaller constellation size, fewer feedforward taps are needed in the DFE for the performance to approach that when there are an infinite number. With 4 QAM, the effect of error propagation in the DFE was seen to be relatively small, unlike with 16 and 64 QAM, for which it appeard increasingly severe. With 4 QAM the MLSE performed at a level close to that of the DFE-I_1, while with 16 and 64 QAM the near-MLSE was, on the whole, worse than the DFE-I_1 and better than the DFE-I_2.

In chapter 6 the task of channel estimation was looked at for two decision-directed tracking algorithms, the SRK and the SD. Assuming correct decisions on data symbols, mathematical analyses of the performances of the two algorithms were presented, with optimum settings for the adaptation parameters derived, and comparison with experiment was generally found to be good. It was shown that the simpler SD algorithm gave a comparable level of accuracy with the more complex SRK algorithm, though its start-up convergence was slower. It is appropriate to mention here a recent paper [94], published near the end of these investigations, which shows measurements of the channel error produced when modified forms of SD and Kalman-type tracking algorithms are used to provide channel estimation (assuming known data symbols) for a simulated 16 QAM two-path HF transmission link. As in chapter 6, the results of [94] show that there is little difference between the two algorithms. Of interest from [94] is the improvement in performance offered at very high *SNR* when the scheme of [66] is adopted to produce a channel estimator that estimates the channel response vector as a linear function of time (recall section 6.4). However, when optimized for steady-state (i.e. long-term) operation, the modified estimators of [94] do exhibit a very much longer start-up convergence time. Finally, it is also worth mentioning here ref. [95], which is basically a more detailed account of the work in [82], some of which was discussed in chapter 6, and therefore may be of interest to the reader.

In chapter 7 the performance of the receiver was observed for when a decision-directed tracking algorithm, assuming correct decisions, is used to provide adaptation. A new method was presented (the CE method) which calculates the MMSE DFE taps from the channel estimate at a computational cost less than the conventional SRK method of implementing the DFE (as given in [16]). Furthermore, the CE method was shown to be superior in both long-term performance and start-up convergence than the conventional method. The revised SRK algorithm of [16], when used to implement the DFE by way of the conventional method, was shown to be unstable when the adaptation parameter ω is set below a certain threshold, this threshold being dependent on the number of taps in the DFE. The performance of the DFE-II$_1$, implemented by way of the CE method, was shown for the two cases where the channel estimate is produced by the SRK and SD tracking algorithms; the difference was small, with the SD being on the whole inferior to the SRK. In general, the *SNR* degradation of the DFE from mode I$_1$ to mode II$_1$ was lower for the smaller constellation size, higher P_e/ξ and lower fade rate; at f.r.=1.0 Hz, $P_e=10^{-3}$, the degradation ranged from 0.5–2.0 dB for 4–64 QAM. It was also shown how, with the CE method, simple predictions of the performance loss of the DFE-II$_1$ with respect to the DFE-I$_1$ can be made. It was observed that in going from mode I to mode II the near-MLSE generally suffered slightly more than the DFE, when both devices used the same channel estimate. Apart from that, the relative performances of the MLSE/near-MLSE and DFE in mode II were more or less similar to those obtained in mode I. The closeness of the near-MLSE to the DFE-II$_2$ around $P_e \lesssim 10^{-3}$ suggests that employment of the near-MLSE be considered carefully in light of its much greater complexity, and probable higher requirement for accuracy in carrier phase recovery.

In chapter 8 it was shown how the DFE (implemented via the CE method) suffers when receiver decisions are used in the tracking algorithm. The degradation from mode II$_2$ to mode III for $P_e \lesssim 10^{-3}$ was small (~0.5 dB) with 4 QAM. With 16 QAM, however, the effect of error bursts was much more severe, and it appeared that the errors in the DFE-II$_1$ would have to be very low for there to be no significant deterioration in mode III. With 64 QAM, the effect of error bursts made performance virtually useless, the error rate barely being below 10^{-1}. Furthermore, with 64 QAM the receiver very often failed to re-align itself following a difficult period, unlike for 4 and 16 QAM. Therefore it was suggested that 16 is the highest constellation size (out of 4, 16 and 64) that can feasibly be used, assuming there is no special arrangement by which the receiver can re-align itself. It was described how the conventional method of implementing the DFE suffered from some instability problems in mode III due to its basic design. When a 10% PTS scheme was used, the performance of the DFE-III was seen to be close to that of the DFE-II$_2$ for all constellation sizes. When an RTS scheme was used, it was shown that very substantial improvements in performance could be achieved for only a small loss in the useful data rate. The use of an RTS scheme appeared particularly attractive with 16 and 64 QAM, in view of the performances obtained previously. It was also shown how the training periods of an RTS scheme could effectively be replaced by useful data transmission with a smaller constellation size, thus allowing continuous throughput of data, and therefore an increase in the useful data rate (referred to as a VDR scheme).

Of the two- and three-path channels used in this thesis, both of which have equal average energy, the extra diversity path of Channel B ensured that in modes I and II the *SNR* necessary to achieve a given $P_e \lesssim 10^{-3}$ on this channel was lower than that on Channel A. In chapter 8, however, it was noted that an RTS scheme may be effective in reducing this diversity advantage, as it often gave rise to a slightly better performance of the DFE on Channel A.

To end this section we will just briefly mention three recent papers related to the area covered in this thesis, these being published while this thesis was being written. In [96] the transmission of 4 QAM signals over a voiceband HF channel is looked at using the conventional implementation of the adaptive DFE, comparing the performances obtained with the FRLS (i.e. Fast-RLS, or Fast-Kalman) algorithm and the LMS (i.e. least mean square, or steepest descent) algorithm. As expected, the FRLS implementation yields a superior performance to that of the LMS, although resetting schemes are necessary with the FRLS to ensure numerical stability. The channels used in [96] have much longer multipath spreads than those used here, and also the DFE is of a fractional-tap type (see section 2.5), employing a $T/2$-spaced feedforward section. It was found necessary in [96] to employ periodic training to ensure receiver recovery from the effects of severe fading. Reference [97] looks at the performance of different forms of near-MLSE, all of which use a pre-filter, on voiceband telephone and HF channels. It is shown that there are more cost-effective ways of selecting which sequences to keep, and which to discard, relative to methods such as those used in [42]. These are based on the notion of selecting proportionally more sequences from those surviviors with lower costs. An interesting result in [97] is that for the transmission of 16 QAM signals over a simulated three-path HF channel, which has a much longer multipath spread than the three-path channel used in this thesis, the advantage of a near-MLSE over a ZF DFE, operating under conditions equivalent to mode I_2, is still not too large, being about 3 dB in terms of signal-to-noise ratio. Finally, in [98] the performances of a finite-tap MMSE DFE and infinite-tap ZF DFE are compared on various time-invariant telephone channels with 16 QAM signals. It is shown that in general there is no significant difference between the MMSE and ZF DFE's, although when severe amplitude distortion is present the MMSE equalizer does perform noticeably better than the ZF equalizer.

Suggestions for further work

It has been observed that transmission with 4 QAM signals is quite robust, the effect of error bursts not perturbing the detection and tracking process to any large degree. The employment of an RTS/VDR scheme offers hope of reliable transmission with 16 and 64 QAM signals, and maybe larger signal sizes too, as well as improved performance with 4 QAM signals, entailing only a small loss in the useful data rate. An interesting area for further work would be to look for more efficient ways of determining performance deterioration. Ideally, an RTS scheme should be able to detect the occurrance of decision errors; the occurrance of error bursts may therefore be advantageous in this respect, since they enhance the detectability of errors.

It would be interesting to observe how much the performance deteriorates when there is also a Doppler shift present, necessitating the employment of a carrier phase recovery scheme. The channels used in this thesis are "good", in the sense that the multipath spread is kept to a minimum for the given number of resolvable paths. Longer channel vectors generally imply less accurate channel estimates, and more taps in the DFE. It would be interesting to see what performance advantage could be obtained with an RTS/VDR scheme when carrier phase recovery is involved, and more larger multipath spreads are present.

Referring back to chapter 2, it was assumed in this thesis that the bandwidth of the data pulse $a(t)$ is $\lesssim T^{-1}$, such that the receiver filter $W(f)$ can effectively be implemented as a fixed low-pass filter, with cut-off at $\pm 0.5/T$, followed by a T-spaced transversal filter (see section 2.5). A lot of useful data pulses, however, have excess bandwidths lying between $1/T$ and $2/T$ [96], and in some circumstances, therefore, an appropriate implementation of $W(f)$ would then be as a fixed low-pass filter with cut-off at $\pm 1/T$, followed by a $T/2$-spaced transversal filter. The DFE in [96] is implemented with a $T/2$-spaced feedforward filter section. In such instances, the MMSE tap settings of the fractional-tap DFE are related in a more complicated way to the channel vector, or rather vectors, since two are now appropriate, these being shifted in time by $T/2$ secs. with respect to each other. It would be interesting to see what advantage could be achieved over the conventional RLS implementation of the fractional-tap DFE, as employed in [96], by deriving the MMSE taps from the two channel estimates, and what the relative cost of doing so might be. One can also consider a more simplified sub-optimum implementation, in which every second tap in the $T/2$-spaced feedforward section forms the T-spaced feedforward section of the MMSE DFE derived from one channel estimate, the other taps similarly forming the MMSE DFE derived from the other channel estimate, so that in effect we have two MMSE

DFE's, each having a *T*-spaced feedforward section, operating in a kind of parallel configuration.

We have not considered the question of redundant coding in this thesis. The potential for coding gain, particularly where the redundancy is in the form of more levels in the signal constellation size, is a topic for further work.

Finally, the results and ideas expressed in this thesis could also have relevance to other time-varying channels, when there is ISI at the receiver. Troposcatter systems [1], [3], [64], [80], [99]–[101] (see section 1.1) are widely used by the military for beyond-the-horizon communications up to 600 miles [1]. Typical multipath spreads are ~1 μs or less, and fade rates ~10 Hz or less. With a typical baud rate of 10 MHz, troposcatter systems therefore do not experience as large a channel variation in one baud interval as do HF systems, and consequently less powerful tracking algorithms can be used [64]. In fact, the adaptive DFE in a troposcatter system is usually implemented via the conventional method using the SD algorithm. There is considerable interest at the moment in UHF mobile radio (~900 MHz) [102]–[108], for providing digital communication within urban and suburban areas of cities. The mobile radio channel has a discrete multipath structure, the different paths arising primarily from the reflection and scattering of the radio waves by buildings and other obstructions. A typical multipath spread is in the range 1–5 μs [106], [107], and fade rates can be as high as 160 Hz, depending on vehicle speed and direction [105].

APPENDICES

Appendix A

Suppose, in the absence of noise, the signal at the receiver is

$$r(t) = \sum_i s_i y(t-iT) = [\sum_i s_i \delta(t-iT)]\bullet y(t) \tag{A.1}$$

where $y(t)$ is the resultant impulse response seen by the receiver and $\{s_i\}$ are the transmitted data symbols (see fig. 2.3 of chapter 2). The FT of $r(t)$ is

$$R(f) = [\sum_i s_i e^{-j2\pi fiT}]Y(f) \tag{A.2}$$

If $r(t)$ is passed through a filter $e(t)$, of FT $E(f)$, the output $r_e(t)$ and its FT are

$$r_e(t) = [\sum_i s_i \delta(t-iT)]\bullet y(t)\bullet e(t) \tag{A.3}$$

$$R_e(f) = [\sum_i s_i e^{-j2\pi fiT}]Y(f)E(f) \tag{A.4}$$

If $r_e(t)$ is sampled at t_o+kT, the resulting signal $r_{e1}(t)$ and its FT are

$$r_{e1}(t) = r_e(t)\sum_k \delta(t-t_o-kT) = \frac{r_e(t)}{T}\sum_k e^{-j2\pi(t-t_o)k/T} \tag{A.5}$$

$$\begin{aligned} R_{e1}(f) &= \frac{1}{T}\sum_k R_e(f+\frac{k}{T})e^{j2\pi t_o k/T} \\ &= [\sum_i s_i e^{-j2\pi fiT}]\, S(f) \end{aligned} \tag{A.6}$$

where
$$S(f) = \frac{1}{T}\sum_k Y(f+\frac{k}{T})E(f+\frac{k}{T})e^{j2\pi t_o k/T} \tag{A.7}$$

If the sample train signal $r_{e1}(t)$ is a set of sufficient statistics for the estimation of the $\{s_i\}$, then this means it should be possible to reconstruct the original signal $r(t)$ by passing $r_{e1}(t)$ through a linear filter. The sampling process can then be deemed "information lossless". Clearly, from consideration of (A.6), we require that

$$S(f)\neq 0 \quad \text{when} \quad Y(f)\neq 0 \tag{A.8}$$

otherwise it would be impossible to reproduce all the non-zero parts of the spectrum $Y(f)$ by linear filtering. If

$$E(f) = Y^*(f)e^{-j2\pi f t_o} \tag{A.9}$$

then

$$S(f) = e^{-j2\pi f t_o}\frac{1}{T}\sum_k |\,Y(f+\frac{k}{T})|^2 \tag{A.10}$$

and (A.8) holds. Note that (A.9) is <u>not</u> necessarily the only solution for $E(f)$ such that (A.8) holds.

Appendix B

Consider the N-component complex vector random variable $\underline{z}$,

$$\underline{z} = \underline{x}+j\underline{y} = [x_1 \ x_2 \ldots x_N]^{\mathrm{t}}+j[y_1 \ y_2 \ldots y_N]^{\mathrm{t}} \tag{B.1}$$

where each of the x_i, y_i, $1 \leq i \leq N$, is a real random variable. Vector $\underline{z}$ can alternatively be represented by the $2N$-component real vector random variable $\underline{c}$,

$$\underline{c}^{\mathrm{t}} = [\underline{x}^{\mathrm{t}} \ \underline{y}^{\mathrm{t}}] = [x_1 \ x_2 \ldots x_N \ y_1 \ y_2 \ldots y_N] \tag{B.2}$$

If $\underline{z}$ is a zero mean complex gaussian vector random variable, then the probability distribution function is

$$p_{\underline{z}}(x_1, \ldots, x_N, y_1, \ldots, y_N) = (2\pi)^{-N}(\det\mathrm{Q})^{-1/2} e^{\{-\frac{1}{2}\underline{c}^{\mathrm{t}}\mathrm{Q}^{-1}\underline{c}\}} \tag{B.3}$$

where
$$\mathrm{Q} = \mathrm{E}[\underline{c} \ \underline{c}^{\mathrm{t}}] \tag{B.4}$$

The characteristic function is

$$\Psi_{\underline{z}}(\underline{\nu}) = \mathrm{E}[e^{j\underline{\nu}^{\mathrm{t}}\underline{c}}] \tag{B.5}$$

where
$$\underline{\nu}^{\mathrm{t}} = [\nu_{x1} \ \nu_{x2} \ldots \nu_{xN} \ \nu_{y1} \ \nu_{y2} \ldots \nu_{yN}] \tag{B.6}$$

For a zero mean complex gaussian vector random variable this becomes

$$\Psi_{\underline{z}}(\underline{\nu}) = e^{\{-\frac{1}{2}\underline{\nu}^{\mathrm{t}}\mathrm{Q}\underline{\nu}\}} \tag{B.7}$$

The quantity $\underline{c}^{\mathrm{t}}\mathrm{Q}^{-1}\underline{c}$ can be written as

$$\underline{c}^{\mathrm{t}}\mathrm{Q}^{-1}\underline{c} = \underline{c}^{\mathrm{t}}\underline{q} = \sum_{m=1}^{N} x_m q_{xm} + \sum_{m=1}^{N} y_m q_{ym} \tag{B.8}$$

where
$$\underline{q} = [q_{x1} \ldots q_{xN} \ q_{y1} \ldots q_{yN}]^{\mathrm{t}} = \mathrm{Q}^{-1}\underline{c} \tag{B.9}$$

From (B.4) and (B.9) one gets

$$\sum_{k=1}^{N}\{\mathrm{E}[x_m x_k]q_{xk}+\mathrm{E}[x_m y_k]q_{yk}\} = x_m \qquad \text{for } 1 \leq m \leq N \tag{B.10}$$

$$\sum_{k=1}^{N}\{\mathrm{E}[y_m x_k]q_{xk}+\mathrm{E}[y_m y_k]q_{yk}\} = y_m \qquad \text{for } 1\leq m\leq N \tag{B.11}$$

Suppose now that under consideration is a stationary complex zero mean gaussian random process $z(t)=x(t)+jy(t)$, where $x(t)$ and $y(t)$ are stationary real zero mean gaussian random processes. Imagine that $z(t)$ consists of an infinite number of samples over a period of observation $\mathfrak{T}$. Then (B.10) and (B.11) can be written in equivalent form for a random process as

$$\int_{\mathfrak{T}}\{r_x(t-u)q_x(u)+r_{xy}(t-u)q_y(u)\}\ du = x(t) \qquad \text{for } t\in\mathfrak{T} \tag{B.12}$$

$$\int_{\mathfrak{T}}\{r_{yx}(t-u)q_x(u)+r_y(t-u)q_y(u)\}\ du = y(t) \qquad \text{for } t\in\mathfrak{T} \tag{B.13}$$

where

$$r_x(\tau) = \mathrm{E}[x(t+\tau)x(t)] \tag{B.14}$$

$$r_{xy}(\tau) = \mathrm{E}[x(t+\tau)y(t)] \tag{B.15}$$

$$r_y(\tau) = \mathrm{E}[y(t+\tau)y(t)] \tag{B.16}$$

$$r_{yx}(\tau) = \mathrm{E}[y(t+\tau)x(t)] \tag{B.17}$$

The equivalent form of the RHS of (B.8) is

$$\int_{\mathfrak{T}}\{x(t)q_x(t)+y(t)q_y(t)\}\ dt \tag{B.18}$$

Thus for the probability distribution function and characteristic function of $z(t)$ one can write

$$p_z(x(t),\ y(t);\ t\in\mathfrak{T}) \sim e^{\{-\frac{1}{2}\int_{\mathfrak{T}}[x(t)q_x(t)+y(t)q_y(t)]\ dt\}} \tag{B.19}$$

$$\Psi_z(\nu_x(t),\ \nu_y(t);\ t\in\mathfrak{T}) \sim e^{\{-\frac{1}{2}\int_{\mathfrak{T}}[\nu_x(t)u_x(t)+\nu_y(t)u_y(t)]\ dt\}} \tag{B.20}$$

where

$$u_x(t) = \int_{\mathfrak{T}}\{r_x(t-u)\nu_x(u)+r_{xy}(t-u)\nu_y(u)\}\ du \tag{B.21}$$

$$u_y(t) = \int_{\mathfrak{T}}\{r_{yx}(t-u)\nu_x(u)+r_y(t-u)\nu_y(u)\}\ du \tag{B.22}$$

If the period of observation $\mathfrak{T}$ is long enough so that boundary conditions are insignificant, then taking FT's of both sides of (B.12) and (B.13) yields

$$R_x(f)Q_x(f)+R_{xy}(f)Q_y(f) = X(f) \tag{B.23}$$

$$R_{yx}(f)Q_x(f)+R_y(f)Q_y(f) = Y(f) \tag{B.24}$$

Solving for $Q_x(f)$ and $Q_y(f)$,

$$Q_x(f) = \frac{X(f)R_y(f)-Y(f)R_{xy}(f)}{R_x(f)R_y(f)-R_{xy}(f)R_{yx}(f)} \tag{B.25}$$

$$Q_y(f) = \frac{Y(f)R_x(f)-X(f)R_{yx}(f)}{R_x(f)R_y(f)-R_{xy}(f)R_{yx}(f)} \tag{B.26}$$

Further, if $x(t)$ and $y(t)$ are statistically independent random processes, i.e.

$$r_{xy}(\tau) = r_{yx}(\tau) = 0 \tag{B.27}$$

then

$$p_z(x(t),\ y(t);\ t\in\Im) \sim e^{\{-\frac{1}{2}\int_\Im\int_\Im[x(t)r_x^{-1}(t-u)x(u)+y(t)r_y^{-1}(t-u)y(u)]\ du\ dt\}} \tag{B.28}$$

where
$$r_x(\tau)\bullet r_x^{-1}(\tau) = \delta(\tau) \tag{B.29}$$

$$r_y(\tau)\bullet r_y^{-1}(\tau) = \delta(\tau) \tag{B.30}$$

and $\Psi_z(\nu_x(t),\ \nu_y(t);\ t\in\Im) \sim$

$$e^{\{-\frac{1}{2}\int_\Im\int_\Im[\nu_x(t)r_x(t-u)\nu_x(u)+\nu_y(t)r_y(t-u)\nu_y(u)]\ du\ dt\}} \tag{B.31}$$

Appendix C

At time kT we have the input vector $\underline{x}_k$, the desired value z_k, and want to update the estimated vector $\underline{c}_{k-1}$ to $\underline{c}_k$. The SRK algorithm [16] proceeds as follows, with $\{x_i\}$, $\{c_i\}$, $\{d_i\}$ and $\{\mu_{ni}\}$ representing the elements of $\underline{x}_k$, $\underline{c}_{k-1}$, diagonal of D_{k-1} and upper triangular portion (excluding diagonal) of U_{k-1} respectively. The number of equivalent real computations involved for a particular stage are given on the right hand side of the page, and the "Δ" markings on the left hand side have relevance to the discussion in section 7.3 of chapter 7. The computational steps given here differ only very slightly from those given in [16], the end result, of course, being the same.

Note: A multiplication of two complex numbers is assumed to require 4 real multiplications and 2 real additions. An addition of two complex numbers is assumed to require 2 real additions.

```
        $\epsilon_k = z_k - \sum_{i=1}^{M} x_i c_i$                     $4M\times$, $4M+$
        $f_1 = x_1^*$
        $g_1 = d_1 f_1$                                           $2\times$
        $\alpha_1 = \eta + g_1 f_1^*$                                  $2\times$, $2+$
        For $i=2$, $M$
            $f_i = \sum_{n=1}^{i-1} \mu_{ni} x_n^* + x_i^*$          $4(i-1)\times$, $4(i-1)+$
            $g_i = d_i f_i$                                       $2\times$
            $\alpha_i = \alpha_{i-1} + g_i f_i^*$                      $2\times$, $2+$
        Next $i$
Δ       $\kappa = \frac{1}{\alpha_1}$                                      $1\div$
Δ       $d_1 = d_1 h_\omega \eta \kappa$    (where $h_\omega = \frac{1}{\omega}$)            $3\times$
        For $i=2$, $M$
Δ           $\beta_0 = \alpha_{i-1}$
            $\lambda = f_i \kappa$                                    $2\times$
```

```
△       $\kappa=\frac{1}{\alpha_i}$                                  $1\div$
        $d_i=d_i h_\omega \beta_0 \kappa$                            $3\times$
        For $n=1,\ i-1$                            ┐
            $\beta_1=\mu_{ni}$                     │
            $\mu_{ni}=\beta_1-\mathbf{g}_n^*\lambda$                  ├─   $8(i-1)\times,\ 8(i-1)+$
            $\mathbf{g}_n=\mathbf{g}_n+\mathbf{g}_i\beta_1^*$        │
        Next $n$                                   ┘
    Next $i$
△   $e=\epsilon_k\kappa$                                            $2\times$
    For $i=1,\ M$                                  ┐
        $c_i=c_i+\mathbf{g}_i e$                                     ├─   $4M\times,\ 4M+$
    Next $i$                                       ┘
```

All quantities are complex except ω, η, h_ω, κ, β_0, $\{d_i\}$ $(1\leq i\leq M)$ and $\{\alpha_i\}$ $(1\leq i\leq M)$. As far as storage is concerned, the algorithm requires $M^2+9M+14$ equivalent real variable locations, a complex variable being assumed to take the storage of two real variables. Note that the lower triangular portion and diagonal elements of $\mathbf{U}_k$ are not used (the diagonal elements of $\mathbf{U}_k$ are unity). The initial start-up values of $\{d_i\}$ and $\{\mu_{ni}\}$ are

$$d_i = 1.0 \quad \text{for} \quad i=1,\ 2,\ \ldots\ ,\ M$$

$$\mu_{ni} = 0.0+j0.0 \quad \text{for} \quad n=1,\ 2,\ \ldots\ ,\ M-1 \ \text{ and } \ i=n+1,\ n+2,\ \ldots\ ,\ M$$

Totals:

Equivalent real multiplications: $6M^2+11M$

Equivalent real additions: $6M^2+4M$

Real reciprocals: M

Exact computational requirements for the algorithm may be slightly different from those listed above for various signal processors.

Appendix D

Consider the complex variable

$$x+jy = re^{j\theta} = \sqrt{x^2+y^2}\; e^{j\tan^{-1}(y/x)} \tag{D.1}$$

where x and y are statistically independent zero mean gaussian random variables of unit variance. The joint probability distribution function (p.d.f.) of (r, θ) is obtained by comparing it with the joint p.d.f. of (x, y) as below, remembering that $dxdy \equiv rdrd\theta$.

$$\begin{aligned} p_{x,y}(\alpha,\, \beta)\; d\alpha\; d\beta &= \frac{1}{2\pi} e^{-(\alpha^2+\beta^2)/2} d\alpha\; d\beta \\ &= \frac{1}{2\pi} e^{-\gamma^2/2} \gamma\; d\gamma\; d\phi = p_{r,\theta}(\gamma,\, \phi)\; d\gamma\; d\phi \end{aligned} \tag{D.2}$$

where $\gamma^2 = \alpha^2+\beta^2$. Hence

$$p_{r,\theta}(\gamma,\, \phi) = \frac{\gamma}{2\pi} e^{-\gamma^2/2} \tag{D.3}$$

Eq. (D.3) clearly shows that r and θ are statistically independent. Therefore

$$p_r(\gamma) = \gamma e^{-\gamma^2/2} \quad \text{for} \quad \gamma \geq 0 \tag{D.4}$$

$$p_\theta(\phi) = \frac{1}{2\pi} \quad \text{for} \quad 0 \leq \phi < 2\pi \tag{D.5}$$

The function $p_r(\gamma)$ is a Rayleigh distribution. Eqs. (D.4) and (D.5) imply that the complex gaussian variable $(x+jy)$ can be obtained from a Rayleigh distributed random variable and a uniformly distributed random variable.

Consider now the independent real random variables z_1 and z_2, each having a uniform p.d.f. between 0 and 1 and a zero p.d.f. elsewhere. Then the variable defined as

$$r' = \sqrt{-2\, \ln(z_1)} \tag{D.6}$$

can easily be shown to have the Rayleigh p.d.f.

$$p_{r'}(\gamma) = \gamma e^{-\gamma^2/2} \quad \text{for} \quad \gamma \geq 0 \tag{D.7}$$

If we define the variable θ' as

$$\theta' = 2\pi z_2 \tag{D.8}$$

$$\Rightarrow \qquad p_{\theta'}(\phi) = \frac{1}{2\pi} \quad \text{for} \quad 0 \leq \phi < 2\pi \tag{D.9}$$

then $r'e^{j\theta'}$ gives us the complex gaussian random variable we are seeking. The variance is 2, but can be changed to any desired value by simply multiplying by an appropriate constant.

Appendix E

Consider the general case of threshold detecting data symbol s_k from $\tilde{s}_k$, which is given as

$$\tilde{s}_k = s_k + \sum_i s_{k-i} b_i + n \tag{E.1}$$

where the second term on the RHS of (E.1) is interference from both past and future data symbols, and also the present. The quantity n is a random gaussian noise variable of variance σ^2. The quantities s_k, b_i and n are in general complex, and we split them into their component real and imaginary parts as below.

$$s_k = s_{k1} + j s_{k2} \tag{E.2}$$

$$b_i = b_{i1} + j b_{i2} \tag{E.3}$$

$$n = n_1 + j n_2 \tag{E.4}$$

Each of the symbols $s_{k1}$and s_{k2} are equally likely to take on any one of the L integer values

$$\alpha_i = 2i - L - 1 \qquad \text{for} \quad 1 \le i \le L \tag{E.5}$$

where L is an even integer, and they do so independent of each other. Successive data symbols s_k are also independent of each other. The noise variables n_1 and n_2 are statistically independent zero mean gaussian random variables of variance $\sigma^2/2$. Splitting (E.1) into its component real and imaginary parts,

$$\text{Re}[\tilde{s}_k] = s_{k1} + \sum_i (s_{k-i,1} b_{i1} - s_{k-i,2} b_{i2}) + n_1 \tag{E.6}$$

$$\text{Im}[\tilde{s}_k] = s_{k2} + \sum_i (s_{k-i,1} b_{i2} + s_{k-i,2} b_{i1}) + n_2 \tag{E.7}$$

The real and imaginary components of the detected symbol $\hat{s}_k = \hat{s}_{k1} + j\hat{s}_{k2}$ are determined from threshold detection of (E.6) and (E.7) respectively. Define

$$\left.\begin{aligned} z_{i1} &= s_{k-i,1} b_{i1} & \qquad u_{i1} &= s_{k-i,2} b_{i1} \\ z_{i2} &= s_{k-i,1} b_{i2} & \qquad u_{i2} &= s_{k-i,2} b_{i2} \end{aligned}\right\} \tag{E.8}$$

Therefore

$$\mathrm{Re}[\tilde{s}_k] = s_{k1}+s_{k1}b_{01}+\sum_{i\neq 0} z_{i1}-\sum_{i} u_{i2}+n_1 \tag{E.9}$$

$$\mathrm{Im}[\tilde{s}_k] = s_{k2}+s_{k2}b_{01}+\sum_{i} z_{i2}+\sum_{i\neq 0} u_{i1}+n_2 \tag{E.10}$$

The ISI in both (E.9) and (E.10) has a maximum value of

$$PD_k = (L-1)\sum_{i}(|b_{i1}|+|b_{i2}|) \tag{E.11}$$

The quantity PD_k is sometimes referred to as the "peak distortion" or the "eye-opening". If PD_k is greater than 1 ("eye" is closed) then no matter how small the noise the error in detection can never be zero.

The p.d.f. for n_1 is

$$p_{n_1}(\gamma) = \frac{1}{\sqrt{\pi}\ \sigma}e^{-\gamma^2/\sigma^2} \tag{E.12}$$

and the p.d.f. for n_2 is identical. The p.d.f. for z_{i1} and u_{i1} is

$$p_{z_{i1}}(\gamma) = p_{u_{i1}}(\gamma) = \frac{1}{L}\sum_{m=1}^{L}\delta(\gamma-\alpha_m|b_{i1}|) \tag{E.13}$$

and the p.d.f.'s for z_{i2} and u_{i2} are identical but with b_{i1} replaced by b_{i2}. Let

$$B = \sum_{i\neq 0} z_{i1}-\sum_{i} u_{i2} \tag{E.14}$$

The probability of error, P_{e1}, in the detection of real symbol $\hat{s}_{k1}$ is

$$\begin{aligned} P_{e1} &= \frac{1}{L}\sum_{i=2}^{L-1}\{\Pr[B+n_1>1-\alpha_i b_{01}]+\Pr[B+n_1<-1-\alpha_i b_{01}]\} \\ &\qquad +\frac{1}{L}\Pr[B+n_1<-1-(L-1)b_{01}]+\frac{1}{L}\Pr[B+n_1>1+(L-1)b_{01}] \\ &= \frac{2}{L}\left\{\sum_{i=2}^{L/2}\Pr[B+n_1>1-\alpha_i b_{01}]+\Pr[B+n_1>1+\alpha_i b_{01}]\right\} \\ &\qquad + \frac{2}{L}\Pr[B+n_1>1+(L-1)b_{01}] \end{aligned} \tag{E.15}$$

where we have used the fact that the p.d.f. of $B+n_1$ is symmetrical; that is, $\Pr[B+n_1>x]=\Pr[B+n_1<-x]$. The probability of error, P_{e2}, in the detection of real symbol

$\hat{s}_{k2}$ is equal to P_{e1} because the interference terms in (E.9) and (E.10) are statistically identical. The overall probability of error, P_{ek}, in the detection of complex data symbol $\hat{s}_k$ is

$$P_{ek} = \Pr[(\text{Error in } \hat{s}_{k1})\bigcup(\text{Error in } \hat{s}_{k2})] \leq P_{e1}+P_{e2} = 2P_{e1} \tag{E.16}$$

Note: For no interference from data symbols, the detection of $\hat{s}_{k1}$ is independent from the detection of $\hat{s}_{k2}$, so that P_{ek} can be written exactly as $P_{ek}=1-(1-P_{e1})(1-P_{e2})\approx 2P_{e1}$ for P_{e1} small.

We will now bound P_{e1} by bounding $\Pr[B+n_1>x]$ along the lines suggested in [79]. Let us first write B in a slightly different form as

$$B = \sum_{i=1}^{J}\beta_i x_i \tag{E.17}$$

where the β_i, $1\leq i\leq J$, are statistically independent binary random variables, equally likely to take on one of the two values ±1. The x_i, $1\leq i\leq J$, are positive multiplying coefficients. It is always possible to write B in this way if $\log_2 L$ is an integer (which it is for $L^2=4$, 16 and 64) because then an L-level random variable $s_{k-i,1}$ can be interpreted as being composed of $\log_2 L$ independent binary variables having the respective ranges $\pm 2^i$, $i=0, 1, \ldots, (\log_2 L-1)$.

A receiver, which sees a $(g+1)$-length sampled channel impulse response and employs a DFE(N,g), is unlikely to get a perfect estimate of the true feedback tap coefficients (necessary for ISI cancellation) from its tracking algorithm. Therefore from (2.132) we can see that

$$J \leq \{2(N+g)-1\}\log_2 L \quad \text{for DFE}(N,g) \tag{E.18}$$

Let the $\{x_i\}$ in (E.17) be arranged in non-increasing order, i.e. $x_{i+1}\leq x_i$. Denoting as $p_B(\gamma)$ the p.d.f. of B, and noting that the p.d.f. of $B+n_1$ is the convolution of $p_{n_1}(\gamma)$ and $p_B(\gamma)$,

$$\Pr[B+n_1>x] = \int_{x}^{\infty}\int_{-\infty}^{\infty} p_{n_1}(u-\gamma)p_B(\gamma)\, d\gamma\, du \tag{E.19}$$

Define the sequence a_m, $-1\leq m\leq J$, as

$$a_{-1} = -(a_J+\delta) \quad \text{with} \quad 0<\delta\ll a_J \tag{E.20}$$

$$a_0 = 0 \tag{E.21}$$

$$a_m = \sum_{i=1}^{m} x_i \quad \text{for} \quad 1 \leq m \leq J \tag{E.22}$$

where it will be noticed that $-a_J \leq B \leq a_J$. Define the probability $\Pr[B : a_m]$ as

$$\Pr[B : a_m] = \Pr[B > a_m] + \tfrac{1}{2}\Pr[B = a_m] \tag{E.23}$$

Now, for $0 \leq m \leq J$,

$$\Pr[B : a_m] = \sum_{\beta_1} \cdots \sum_{\beta_m} \Pr[\beta_1, \ldots, \beta_m] \Pr[\sum_{i=1}^{m} \beta_i x_i + W_m : a_m \,/\, \beta_1, \ldots, \beta_m] \tag{E.24}$$

where

$$W_m = \sum_{i=m+1}^{J} \beta_i x_i \tag{E.25}$$

Now $\Pr[\beta_1, \ldots, \beta_m] = 2^{-m}$, so

$$\Pr[B : a_m] = 2^{-m}(\Pr[W_m : 0] + \epsilon_m) \tag{E.26}$$

where

$$\epsilon_m = \sum_{i=1}^{m} \Pr[W_m : 2x_i] + \frac{1}{2!} \sum_{i=1}^{m} \sum_{\substack{p=1 \\ p \neq i}}^{m} \Pr[W_m : 2x_i + 2x_p] + \cdots + \Pr[W_m : 2a_m] \tag{E.27}$$

$(\epsilon_0 = \epsilon_J = 0)$

Since $\Pr[W_m : 0] = \frac{1}{2}$ we have

$$\Pr[B : a_m] = 2^{-(m+1)} + 2^{-m}\epsilon_m \quad \text{for} \quad 0 \leq m \leq J \tag{E.28}$$

Subtracting $\Pr[B : a_{m+1}]$ from $\Pr[B : a_m]$ gives us

$$\begin{aligned} \Pr[a_{m+1} : B : a_m] &= -\tfrac{1}{2}\Pr[B = a_{m+1}] + \Pr[a_{m+1} \geq B > a_m] + \tfrac{1}{2}\Pr[B = a_m] \\ &= 2^{-(m+2)} + 2^{-(m+1)}(2\epsilon_m - \epsilon_{m+1}) \quad \text{for} \quad 0 \leq m \leq J-1 \end{aligned} \tag{E.29}$$

Now from (E.12) and (E.19),

$$\Pr[B + n_1 > x] = \int_{-\infty}^{\infty} p_B(\gamma) \int_{x}^{\infty} \frac{1}{\sqrt{\pi}\,\sigma} e^{-(u-\gamma)^2/\sigma^2} \, du \, d\gamma$$

$$= \int_{-\infty}^{\infty} p_B(\gamma)\tfrac{1}{2}\mathrm{erfc}(\tfrac{x-\gamma}{\sigma})\ d\gamma$$

$$= \sum_{i=0}^{J} \int_{a_{i-1}}^{a_i} p_B(\gamma)\tfrac{1}{2}\mathrm{erfc}(\tfrac{x-\gamma}{\sigma})\ d\gamma \tag{E.30}$$

where

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-t^2} dt \tag{E.31}$$

$\Pr[a_i : B : a_{i-1}]$ can be written in terms of $p_B(\gamma)$ as

$$\Pr[a_i : B : a_{i-1}] = -\tfrac{1}{2}p_B(a_i)\ d\gamma + \int_{a_{i-1}}^{a_i} p_B(\gamma)\ d\gamma + \tfrac{1}{2}p_B(a_{i-1})\ d\gamma \tag{E.32}$$

the range in the integrand being $a_{i-1}<\gamma\leq a_i$. Using (E.32), and the fact that $\mathrm{erfc}(\frac{x-\gamma}{\sigma})$ is a non-decreasing function of γ, we can bound (E.30) as

$$\sum_{i=0}^{J}\Pr[a_i : B : a_{i-1}]q_{i-1} + \tfrac{1}{2}\Pr[B=a_J]q_J \leq \Pr[B+n_1>x]$$

$$\leq \sum_{i=0}^{J}\Pr[a_i : B : a_{i-1}]q_i + \tfrac{1}{2}\Pr[B=a_J]q_J \tag{E.33}$$

where

$$q_i = \tfrac{1}{2}\mathrm{erfc}(\tfrac{x-a_i}{\sigma}) \tag{E.34}$$

Substituting $\Pr[B=a_J]=2^{-J}$, $\Pr[a_0 : B : a_{-1}]=\frac{1}{2}$ and (E.29) we get

$$\sum_{i=1}^{J}[2^{-(i+1)}+2^{-i}(2\epsilon_{i-1}-\epsilon_i)]\ q_{i-1}+\tfrac{1}{2}q_{-1}+2^{-(J+1)}q_J \leq \Pr[B+n_1>x]$$

$$\leq \sum_{i=1}^{J}[2^{-(i+1)}+2^{-i}(2\epsilon_{i-1}-\epsilon_i)]\ q_i+\tfrac{1}{2}q_0+2^{-(J+1)}q_J \tag{E.35}$$

Looking at the lower bound (LB) and upper bound (UB) of (E.35), and using the fact that $\epsilon_0=\epsilon_J=0$,

$$\mathrm{LB} = \sum_{i=0}^{J-1}[2^{-(i+2)}+2^{-(i+1)}(2\epsilon_i-\epsilon_{i+1})]\ q_i+\tfrac{1}{2}q_{-1}+2^{-(J+1)}q_J$$

$$= \tfrac{1}{2}(\bar{P}+2^{-J}q_J+q_{-1}+\Delta_{\mathrm{L}}) \tag{E.36}$$

where

$$\bar{P} = \sum_{i=0}^{J-1}2^{-(i+1)}q_i \tag{E.37}$$

$$\Delta_{\mathrm{L}} = \sum_{i=0}^{J-1}2^{-i}(2\epsilon_i-\epsilon_{i+1})q_i = \sum_{i=0}^{J-2}2^{-i}\epsilon_{i+1}(q_{i+1}-q_i) \geq 0 \tag{E.38}$$

$$\mathrm{UB} = \bar{P}+2^{-J}q_J+\Delta_{\mathrm{U}} \tag{E.39}$$

where $$\Delta_{\mathrm{U}} = \sum_{i=1}^{J} 2^{-i}(2\epsilon_{i-1}-\epsilon_i)q_i = \sum_{i=0}^{J-2} 2^{-(i+1)}\epsilon_{i+1}(q_{i+2}-q_{i+1}) \geq 0 \tag{E.40}$$

It is demonstrated in [79] that we can assume Δ_{L}, $\Delta_{\mathrm{U}} \ll \bar{P}$. Hence we can approximate (E.35) to

$$\tfrac{1}{2}(\bar{P}+2^{-J}q_J+q_{-1}) \leq \Pr[B+n_1>x] \leq \bar{P}+2^{-J}q_J \tag{E.41}$$

Note that the lower bound is a true lower bound, whereas the upper one is an approximation (which is less) because of our assumption about Δ_{U}.

For $J=0$ (no ISI) we have $\bar{P}=0$ and $q_{-1}=q_0$, and the lower and upper bounds become equal and give the exact value for the probability; since $b_{01}=0$ and $x=1$, the RHS of (E.16) becomes $2(1-\frac{1}{L})\mathrm{erfc}(\frac{1}{\sigma})$. For $J>0$, as a_J increases then $q_{-1}\rightarrow 0$ and the lower bound tends to half the value of the upper bound.

We have adopted the upper bound, $\bar{P}+2^{-J}q_J$, of (E.41) in this thesis, and evaluate it for each term in (E.15) to provide an estimate of the probability of error (E.16) for a DFE(N,g).

The bounds derived in [79] differ from (E.41) in that the q_{-1} term in the lower bound is absent. This is because in [79] they have defined $a_{-1}=-\infty$, which means $q_{-1}=0$. The lower bound in (E.41) is therefore tighter than that in [79], the difference being greater the smaller the peak distortion a_J is.

Appendix F

We wish to find the expectation of erfc($\sqrt{x}$), where x is the sum of the squares of $2(g+1)$ real independent zero mean gaussian random variables, each of variance $\lambda/2$. Since the p.d.f. of the sum of two independent random variables is the convolution of their respective p.d.f.'s, it can be shown that the p.d.f. of x is

$$p_x(\gamma) = \frac{\gamma^g e^{-\gamma/\lambda}}{\lambda^{g+1} g!} \quad \text{for} \quad \gamma \geq 0 \tag{F.1}$$

Let A denote the quantity we seek, i.e.

$$A = \mathrm{E}[\mathrm{erfc}(\sqrt{x})] = \int_0^\infty \mathrm{erfc}(\sqrt{\gamma}) \frac{\gamma^g e^{-\gamma/\lambda}}{\lambda^{g+1} g!} \, d\gamma \tag{F.2}$$

Now it can be shown that

$$\int_0^\infty e^{-xt} \mathrm{erfc}(\sqrt{t}) \, dt = \frac{1}{x}\left(1 - \frac{1}{\sqrt{1+x}}\right) \tag{F.3}$$

Denoting the LHS of (F.3) by Q,

$$\frac{d^n Q}{dx^n} = \int_0^\infty (-1)^n t^n e^{-xt} \mathrm{erfc}(\sqrt{t}) \, dt \tag{F.4}$$

from which it can be seen that

$$A = \frac{1}{\lambda^{g+1} g!} (-1)^g \left. \frac{d^g Q}{dx^g} \right|_{x=\lambda^{-1}} \tag{F.5}$$

Now

$$\frac{d^n(x^{-1})}{dx^n} = \frac{n!\,(-1)^n}{x^{n+1}} \tag{F.6}$$

and using the formula

$$\frac{d^n(pq)}{dx^n} = \frac{d^n p}{dx^n} q + {}^nC_1 \frac{d^{n-1} p}{dx^{n-1}} \frac{dq}{dx} + \cdots + p \frac{d^n q}{dx^n} \quad \text{for} \quad n \geq 1 \tag{F.7}$$

we obtain, using $p=1/x$ and $q=1/\sqrt{1+x}$,

$$\frac{d^n (x\sqrt{1+x})^{-1}}{dx^n} = \sum_{k=0}^{n} {}^nC_k \, \frac{(n-k)!}{x^{n-k+1}} \, 2(-1)^{n+1} (1+x)^{-k-\frac{1}{2}} \prod_{i=0}^{k} (i - \tfrac{1}{2}) \tag{F.8}$$

Thus
$$\frac{d^n Q}{dx^n} = \frac{n!\,(-1)^n}{x^{n+1}}\Big(1-\frac{1}{\sqrt{1+x}}\sum_{k=0}^{n}\Big(\frac{x}{1+x}\Big)^k \frac{(-2)}{k!}\prod_{i=0}^{k}(i-\tfrac{1}{2})\Big) \tag{F.9}$$

It can be shown that

$$\Gamma(k+\tfrac{1}{2}) = \sqrt{\pi}\,(-2)\prod_{i=0}^{k}(i-\tfrac{1}{2}) \tag{F.10}$$

where $\Gamma(.)$ is the gamma function, defined as

$$\Gamma(u) = \int_0^\infty x^{u-1}e^{-x}\,dx \tag{F.11}$$

which converges for $u>0$. $\Gamma(1)=1$ and $\Gamma(\frac{1}{2})=\sqrt{\pi}$, and from integration by parts it can be shown that

$$\Gamma(u) = \frac{\Gamma(u+1)}{u} \tag{F.12}$$

leading to the more general expression

$$\Gamma(u+k) = \Gamma(u)\prod_{i=0}^{k-1}(u+i) \qquad \text{for} \quad k=1,\,2,\,3\,\ldots \tag{F.13}$$

Hence

$$A = 1-\frac{1}{\sqrt{\pi}}\sqrt{\frac{\lambda}{1+\lambda}}\sum_{k=0}^{g}\frac{\Gamma(k+\frac{1}{2})}{k!\,(1+\lambda)^k} \tag{F.14}$$

Consider the expectation of $\frac{1}{x}$.

$$\mathrm{E}[\tfrac{1}{x}] = \int_0^\infty \frac{\gamma^{g-1}e^{-\gamma/\lambda}}{\lambda^{g+1}g!}\,d\gamma = \frac{1}{\lambda g} \qquad \text{for} \quad g>0 \tag{F.15}$$

Appendix G

The following is a description of the near-MLSE scheme referred to as "Detector 2" in [42].

On receipt at time kT of the sample u_k, which is from the output of the feedforward section of a DFE (ideally ZF), the detector holds m survivor-sequences, each of length q, denoted as

$$\{\alpha_i\}_{k-q}^{k-1} = \alpha_{k-q}, \alpha_{k-q+1}, \dots, \alpha_{k-1} \tag{G.1}$$

where α_i is a possible data symbol value. Each survivor has associated with it a cost $|\Gamma_{k-1}|^2$. The detector now expands each survivor into L $(q+1)$-length sequences $\{\alpha_i\}_{k-q}^{k}$, the last component α_k taking on the L real values $2i-L-1$, $1 \leq i \leq L$. Note that α_k is not as yet a possible data symbol value. Thus the detector now holds Lm sequences of length $q+1$. Let us define d_k as

$$d_k = u_k - \sum_{i=0}^{g} \alpha_{k-i} b_i = \nu_k - \alpha_k \tag{G.2}$$

where

$$\nu_k = u_k - \sum_{i=1}^{g} \alpha_{k-i} b_i \tag{G.3}$$

$$b_0 = 1 \tag{G.4}$$

and $\{b_i\}$, $1 \leq i \leq g$, are the feedback taps of the DFE. The detector now selects n $(n \leq L)$ sequences that have originated from a particular survivor, discarding the other $L-n$, such that the absolute values of the real part of d_k are minimum. In other words, the detector selects the n sequences originating from a particular survivor for which the given possible values of α_k are closest to the real part of ν_k (G.3), this being achieved by simple threshold comparison and not by the evaluation of any costs.

This process is then repeated for each q-length survivor being expanded into L $(q+1)$-length sequences for which the last compnent α_k this time takes on the L values $j(2i-L-1)$, $1 \leq i \leq L$, the n selected sequences originating from a particular survivior being chosen such that α_k is now closest to the imaginary part of ν_k.

The detector now has m sets of $2n$ $(q+1)$-length sequences, each set originating from a particular survivor. The detector now forms a set of n^2 sequences from each set of $2n$ sequences by letting the last component α_k take on all possible combinations of the real and imaginary values of α_k in the latter set, thereby making it a possible data symbol value. Thus the detector now has m sets of n^2 $(q+1)$-length sequences. For each sequence the detector evaluates the cost

$$|\Gamma_k|^2 = |\Gamma_{k-1}|^2+|d_k|^2 \tag{G.5}$$

using the real and imaginary parts of d_k already evaluated.

Out of the n^2m sequences it holds, the detector selects that with the smallest cost and takes the value of the first component α_{k-q} of this sequence as the detected value $\hat{s}_{k-q}$ of the transmitted data symbol s_{k-q}. All sequences for which $\alpha_{k-q}\neq\hat{s}_{k-q}$ are now discarded, and the first component of each of the remaining sequences (including that with the smallest cost) is omitted to give a set of q-length sequences $\{\alpha_i\}_{k-q+1}^{k}$. The detector now has to select m sequences from the remaining sequences it holds. The first of the selected sequences is that with the smallest cost. If the remaining sequences contain any originating from the smallest-cost survivor, the detector selects from this group the sequence with the smallest cost and then from all the rest of the sequences (which excludes the two just selected) the $m-2$ sequences with the smallest costs. If the remaining sequences do not include any originating from the smallest-cost survivor, the detector simply selects the $m-1$ sequences with the smallest costs. These m selected sequences now form the new set of q-length survivors $\{\alpha_i\}_{k-q+1}^{k}$. They are stored in the detector, along with their costs $|\Gamma_k|^2$, ready for the next sample u_{k+1} at time $(k+1)T$.

In our investigations we set $n=2$ and $m=16$ whenever we use this near-MLSE scheme.

Points of Note from [42]

(1) In the practical implementation of the detection scheme it is not really necessary to

perform the expansion processes described, these being presented to explain what is done by the detector rather than how it precisely does it.

(2) The selection, where possible, of at least one sequence originating from the smallest-cost survivor is done because it has been found to reduce significantly the average length of error bursts, when the channel introduces severe amplitude distortion into the received data signal.

(3) The discarding of sequences for which $\alpha_{k-q} \neq \hat{s}_{k-q}$ prevents the merging (i.e. becoming the same) of the survivors, since it ensures that if these are all different at the start of transmission then no two or more of them can subsequently become the same.

Appendix H

Given the $N \times N$ lower triangular Toeplitz matrix $\mathbf{L}$, we wish to solve for the N-length vector $\underline{\mathbf{w}}$ from the equation

$$(\mathbf{L}^{\mathrm{t}}+\hat{p}\mathbf{L}^{*-1})\underline{\mathbf{w}} = [1\ 0\ \ldots\ 0]^{\mathrm{t}} \tag{H.1}$$

where $\hat{p}$ is a real positive scalar. Toeplitz matrices belong to the larger class of persymmetric matrices, which have the property that they are symmetric about their main northeast-southwest diagonal [87]. This is equivalent to requiring

$$\mathbf{A} = \mathbf{I}'\mathbf{A}^{\mathrm{t}}\mathbf{I}' \tag{H.2}$$

where $\mathbf{A}$ is an $N \times N$ persymmetric matrix and $\mathbf{I}'$ is an $N \times N$ matrix which is all zero except for unit elements on its main northeast-southwest diagonal ($\mathbf{I}'\mathbf{I}'=\mathbf{I}$, the identity matrix).

The inverse of a persymmetric matrix is also persymmetric. The inverse of a Toeplitz matrix is not in general Toeplitz. The inverse of a lower triangular Toeplitz matrix is, however, Toeplitz. Let us denote the inverse of $\mathbf{L}^*$ as $\mathbf{Q}$, and the first column of $\mathbf{L}$, which characterizes the matrix, by the vector

$$\underline{\mathbf{l}}^{\mathrm{t}} = [l_1\ l_2\ \ldots\ l_N] \tag{H.3}$$

We have

$$\mathbf{L}^*\mathbf{Q} = \mathbf{I} \tag{H.4}$$

Equating corresponding elements in (H.4), with $\{q_{ih}\}$ representing the elements of $\mathbf{Q}$,

$$\sum_{k=1}^{i} l^*_{i-k+1} q_{kh} = \delta_{ih} \qquad \text{for} \quad 1 \leq (i,\, h) \leq N$$

$$\Rightarrow \qquad q_{ih} = (\delta_{ih} - \sum_{k=1}^{i-1} l^*_{i-k+1} q_{kh})/l^*_1 \tag{H.5}$$

Given $\underline{\mathbf{l}}$, (H.5) enables us to calculate the elements of $\mathbf{Q}$ column by column. We can see that $q_{ih}=0$ for $h>i$, which means $\mathbf{Q}$ is also a lower triangular matrix. We can therefore rewrite (H.5) as

$$q_{ih} = (\delta_{ih} - \sum_{k=h}^{i-1} l^*_{i-k+1} q_{kh})/l^*_1 \qquad \text{for} \quad 1 \leq h \leq i \leq N \tag{H.6}$$

For $h \leq i \leq N-1$, element $q_{i+1,h+1}$ is given by

$$q_{i+1,h+1} = (\delta_{i+1,h+1} - \sum_{k=h+1}^{i} l^*_{i-k+2} q_{k,h+1})/l^*_1 = (\delta_{ih} - \sum_{k=h}^{i-1} l^*_{i-k+1} q_{k+1,h+1})/l^*_1 \tag{H.7}$$

Comparing (H.7) with (H.6) one can see that $\mathbf{Q}$ is also a Toeplitz matrix, because each successive column is a down-shifted version of the previous one. Since $\mathbf{L}^{\mathrm{t}}$ is an upper triangular Toeplitz matrix,

$$\mathbf{A} = (\mathbf{L}^{\mathrm{t}} + \hat{\rho}\mathbf{Q}) \tag{H.8}$$

is an $N \times N$ Toeplitz matrix. The structure of $\mathbf{A}$ may be drawn as

$$\mathbf{A} = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{-(N-1)} \\ a_1 & a_0 & \cdots & a_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1} & a_{N-2} & \cdots & a_0 \end{bmatrix} \tag{H.9}$$

Let the vectors $\underline{\mathrm{a}}_k$ and $\underline{\mathrm{a}}_{-k}$ be defined, for $1 \leq k \leq N-1$, as

$$\underline{\mathrm{a}}^{\mathrm{t}}_k = [a_1 \; a_2 \; \ldots \; a_k] \tag{H.10}$$

$$\underline{\mathrm{a}}^{\mathrm{t}}_{-k} = [a_{-1} \; a_{-2} \; \ldots \; a_{-k}] \tag{H.11}$$

and let the matrix $\mathbf{A}_k$ be the $k \times k$ top left-hand corner of $\mathbf{A}$ ($\mathbf{A}_N = \mathbf{A}$). Suppose now that we have the solutions $\underline{\mathrm{u}}_k$ and $\underline{\mathrm{v}}_k$ to the equations

$$\mathbf{A}_k \underline{\mathrm{u}}_k = \underline{\mathrm{a}}_k \tag{H.12}$$

$$\mathbf{A}^{\mathrm{t}}_k \underline{\mathrm{v}}_k = \underline{\mathrm{a}}_{-k} \tag{H.13}$$

which are easy to solve for $k=1$. Eqs. (H.12) and (H.13) bear some similarity to the k^{th} order complex Yule-Walker systems [3], [87], the difference being that the latter assumes matrix $\mathbf{A}$ is Hermetian as well, so that $\underline{\mathrm{a}}_k = \underline{\mathrm{a}}^*_{-k}$. The method to be described here for finding $\underline{\mathrm{w}}$ is a simple extension of the Levinson-Durbin algorithm [3], [87] for solving Yule-Walker systems.

First we need to find the vectors $\underline{u}_{k+1}$ and $\underline{v}_{k+1}$ from $\underline{u}_k$ and $\underline{v}_k$. Vector $\underline{u}_{k+1}$ is obtained by solving

$$\begin{bmatrix} \mathrm{A}_k & \mathrm{I}'_k\underline{a}_{-k} \\ \underline{a}^{\mathrm{t}}_k\mathrm{I}'_k & a_0 \end{bmatrix}\begin{bmatrix} \underline{z}_k \\ \alpha_k \end{bmatrix} = \begin{bmatrix} \underline{a}_k \\ a_{k+1} \end{bmatrix} \tag{H.14}$$

where I'_k is the k×k top left-hand corner of I', and

$$\underline{u}^{\mathrm{t}}_{k+1} = [\underline{z}^{\mathrm{t}}_k\alpha_k] \tag{H.15}$$

Equating corresponding terms in (H.14) leads to

$$\mathrm{A}_k\underline{z}_k + \mathrm{I}'_k\underline{a}_{-k}\alpha_k = \underline{a}_k \tag{H.16}$$

$$\underline{a}^{\mathrm{t}}_k\mathrm{I}'_k\underline{z}_k + a_0\alpha_k = a_{k+1} \tag{H.17}$$

Using the relations $\mathrm{A}_k^{-1}\mathrm{I}'_k = \mathrm{I}'_k(\mathrm{A}^{\mathrm{t}}_k)^{-1}$ and $\mathrm{I}'_k\mathrm{I}'_k = \mathrm{I}_k$, where I_k is the k×k identity matrix, we solve (H.16) and (H.17) to get

$$\underline{z}_k = \underline{u}_k - \alpha_k\mathrm{I}'_k\underline{v}_k \tag{H.18}$$

$$\alpha_k = (a_{k+1} - \underline{a}^{\mathrm{t}}_k\mathrm{I}'_k\underline{u}_k)/(a_0 - \underline{a}^{\mathrm{t}}_k\underline{v}_k) \tag{H.19}$$

In a similar manner $\underline{v}_{k+1}$ is obtained by solving

$$\begin{bmatrix} \mathrm{A}^{\mathrm{t}}_k & \mathrm{I}'_k\underline{a}_k \\ \underline{a}^{\mathrm{t}}_{-k}\mathrm{I}'_k & a_0 \end{bmatrix}\begin{bmatrix} \underline{r}_k \\ \beta_k \end{bmatrix} = \begin{bmatrix} \underline{a}_{-k} \\ a_{-(k+1)} \end{bmatrix} \tag{H.20}$$

where

$$\underline{v}^{\mathrm{t}}_{k+1} = [\underline{r}^{\mathrm{t}}_k\beta_k] \tag{H.21}$$

We thus obtain for $\underline{r}_k$ and β_k,

$$\underline{r}_k = \underline{v}_k - \beta_k\mathrm{I}'_k\underline{u}_k \tag{H.22}$$

$$\beta_k = (a_{-(k+1)} - \underline{a}^{\mathrm{t}}_{-k}\mathrm{I}'_k\underline{v}_k)/(a_0 - \underline{a}^{\mathrm{t}}_{-k}\underline{u}_k) \tag{H.23}$$

From (H.12) and (H.13) it is easy to show that

$$\underline{a}^{\mathrm{t}}_{-k}\underline{u}_k = \underline{a}^{\mathrm{t}}_k\underline{v}_k \tag{H.24}$$

which implies that the denominators in (H.19) and (H.23) are equal. Denoting this denominator as p_k, we note that its computation can be expressed recursively as

$$\begin{aligned} p_k &= a_0 - \underline{\mathbf{a}}_k^{\mathrm{t}} \underline{\mathbf{v}}_k = a_0 - [\underline{\mathbf{a}}_{k-1}^{\mathrm{t}} a_k] \begin{bmatrix} \underline{\mathbf{v}}_{k-1} - \beta_{k-1} \mathbf{I}'_{k-1} \underline{\mathbf{u}}_{k-1} \\ \beta_{k-1} \end{bmatrix} \\ &= a_0 - \underline{\mathbf{a}}_{k-1}^{\mathrm{t}} \underline{\mathbf{v}}_{k-1} + \beta_{k-1} \underline{\mathbf{a}}_{k-1}^{\mathrm{t}} \mathbf{I}'_{k-1} \underline{\mathbf{u}}_{k-1} - \beta_{k-1} a_k \\ &= p_{k-1} + \beta_{k-1}(-\alpha_{k-1} p_{k-1}) \\ &= (1 - \alpha_{k-1} \beta_{k-1})\, p_{k-1} \end{aligned} \tag{H.25}$$

To obtain the vector $\underline{\mathbf{w}}^{\mathrm{t}} = [w_0 \underline{\mathbf{w}}'^{\mathrm{t}}]$, observe that

$$\begin{bmatrix} a_0 & \underline{\mathbf{a}}_{-(N-1)}^{\mathrm{t}} \\ \underline{\mathbf{a}}_{N-1} & \mathbf{A}_{N-1} \end{bmatrix} \begin{bmatrix} w_0 \\ \underline{\mathbf{w}}' \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{H.26}$$

Equating corresponding elements in (H.26) gives us the equations

$$a_0 w_0 + \underline{\mathbf{a}}_{-(N-1)}^{\mathrm{t}} \underline{\mathbf{w}}' = 1 \tag{H.27}$$

$$w_0 \underline{\mathbf{a}}_{N-1} + \mathbf{A}_{N-1} \underline{\mathbf{w}}' = \underline{0} \tag{H.28}$$

which we solve to get

$$\underline{\mathbf{w}}' = -w_0 \underline{\mathbf{u}}_{N-1} \tag{H.29}$$

$$w_0 = \frac{1}{a_0 - \underline{\mathbf{a}}_{-(N-1)}^{\mathrm{t}} \underline{\mathbf{u}}_{N-1}} = \frac{1}{p_{N-1}} \tag{H.30}$$

Thus the vector $\underline{\mathbf{w}}^{\mathrm{t}} = [w_0 \underline{\mathbf{w}}'^{\mathrm{t}}]$ can be obtained directly from $\underline{\mathbf{u}}_{N-1}$ and p_{N-1}. (Note that the elements of $\underline{\mathbf{w}}$ run from w_0 to w_{N-1}).

If we are determining the taps of a DFE, as described in section 7.1, then, in the notation of chapter 7, vector $\underline{\mathbf{l}}$ is $\underline{\mathbf{c}}_{k-1,0}$ and $\underline{\mathbf{w}}$ is $\hat{\underline{\mathbf{w}}}_k$. The quantities $\{b_i\}$, $1 \leq i \leq g$, given by

$$b_i = \sum_{h=0}^{g-i} w_h l_{h+i+1} \tag{H.31}$$

are, in the notation of chapter 7, the feedback taps $\{\hat{b}_{ki}\}$, $1 \leq i \leq g$. We will now present an algorithm for the determination of the DFE taps, given the vector $\underline{\mathbf{l}}$ and real scalar $\hat{\rho}$. Note that since the vector $\underline{\mathbf{a}}_{-(N-1)}$ is just the last $(N-1)$ elements of $\underline{\mathbf{l}}$, to save storage we can use the latter in those computational steps that involve the former. Also, we can use the storage elements $\{w_i\}$ in those computational steps that involve the elements of $\underline{\mathbf{u}}_k$. The last $N-g-1$ elements (assuming $N>g$) of $\underline{\mathbf{l}}$ are zero. The number of equivalent real computations involved for a particular stage are given on the right-hand side of the page.

Note: A complex reciprocal is assumed to require 4 real multiplications, 1 real addition and 1 real reciprocal. (See also the note in Appendix C).

(*i*) Determination of a_0 and $\underline{\mathbf{a}}_{N-1}$

```
$\lambda = 1/l_1^*$                                              4×, 1+, 1÷
$a_0 = \hat{\rho}\lambda$                                        2×
For $i=1,\ N-1$
    $a_i = 0.0 + j0.0$
    For $k = \max\{0,\ i-g\},\ i-1$                     ┐
        $a_i = a_i - l^*_{i-k+1} a_k$                   ├─   4 min{i, g}×, 4 min{i, g}+
    Next $k$                                            ┘
    $a_i = a_i\lambda$                                           4×, 2+
Next $i$
$a_0 = a_0 + l_1$                                                2+
```

(*ii*) Determination of $\underline{\mathbf{w}}$ and $\{b_i\}$

```
$\lambda = 1/a_0$                                                4×, 1+, 1÷
$w_1 = a_1\lambda$                                               4×, 2+
$v_1 = l_2\lambda$                                               4×, 2+
$p = a_0$
```

```
$\alpha=w_1$
$\beta=v_1$
For $k=1,\ N-2$
    $p=(1-\alpha\beta)p$                                      $8\times,\ 5+$
    $\alpha=a_{k+1}$
    For $i=1,\ k$
        $\alpha=\alpha-a_i w_{k-i+1}$                         $4k\times,\ 4k+$
    Next $i$
    $\lambda=1/p$                                             $4\times,\ 1+,\ 1\div$
    $\alpha=\alpha\lambda$                                    $4\times,\ 2+$
    $\beta=l_{k+2}$
    For $i=1,\ \min\{k,\ g\}$
        $\beta=\beta-l_{i+1}v_{k-i+1}$                        $4\min\{k,\ g\}\times,\ 4\min\{k,\ g\}+$
    Next $i$
    $\beta=\beta\lambda$                                      $4\times,\ 2+$
    For $i=1,\ k$
        $\lambda=v_{k-i+1}-\beta w_i$
        $w_i=w_i-\alpha v_{k-i+1}$                            $8k\times,\ 8k+$
        $v_{k-i+1}=\lambda$
    Next $i$
    $w_{k+1}=\alpha$
    $v_{k+1}=\beta$
Next $k$
$p=(1-\alpha\beta)p$                                          $8\times,\ 5+$
$w_0=1/p$                                                     $4\times,\ 1+,\ 1\div$
For $i=1,\ N-1$
    $w_i=-w_0 w_i$                                            $4(N-1)\times,\ 2(N-1)+$
Next $i$
```

```
For $i=1,\ g$
    $b_i=0.0+j0.0$
    For $h=0,\ \min\{N-1,\ g-i\}$
        $b_i=b_i+w_h l_{h+i+1}$          $4(g-i+1)\times,\ 4(g-i+1)+$
    Next $h$                               (assuming $N>g$)
Next $i$
```

All quantities are complex except $\hat{p}$. The quantities involved are λ, $\hat{p}$, p, α, β, $\{a_i\}$ $(0\leq i\leq N-1)$, $\{l_i\}$ $(1\leq i\leq g+1)$, $\{v_i\}$ $(1\leq i\leq N-1)$, $\{w_i\}$ $(0\leq i\leq N-1)$ and $\{b_i\}$ $(1\leq i\leq g)$. Therefore the number of equivalent real variable storage locations required is $6N+4g+9$.

Totals:

Equivalent real multiplications: $6N^2+8gN-2g^2+10N-6g-6$

Equivalent real additions: $6N^2+8gN-2g^2-4N-6g+2$

Equivalent real reciprocals: $N+1$

Appendix I

This appendix contains listings of the simulation programs used to produce the results of this thesis. The language is Fortran-77, and the programs were run on a Cyber mainframe. Any variable whose first letter is from I–N should be assumed to be an integer, and a variable with one or two argument(s) is a vector or matrix respectively.

Consider the main program MAIN (given on p. 264). The subroutine VALUES (not detailed) is a routine to provide values to the following variables, which may be in an "interactive" manner or by reading from a separate file containing the values of the variables.

NT: the value of g+1 (NT$\geq$2).

SPEC(I): (complex) the value of the specular component of the Ith path in the channel model (assumed zero in this thesis).

STAND(I): the standard deviation of the real (and imaginary) component of the Ith fading path component in the channel response (assumed to be $1/\sqrt{2(g+1)}$ in this thesis).

DOPPS: the Doppler shift $\times 2\pi/T$ (assumed to be zero in this thesis).

IFADR: the value of $1/(5\sqrt{2}\pi f_r T)$. With T^{-1}=2400, IFADR=216 and 108 implies $f_r \approx 0.5$ and 1.0 Hz respectively.

NSL: the value of L (NSL=2, 4 or 8).

SNR0: the value of SNR (dB).

EPSIL: the value of η/σ_s^2 in the SRK algorithm.

HQUEP: the value of $1/\omega$ in the SRK algorithm.

DELTA: the value of μ' in the SD algorithm.

LANT: the number of feedforward taps in the DFE.

LEMM, MUE: the values of m and n respectively in the near-MLSE scheme of Appendix G.

IDID: the value of the delay in detection, q, of the MLSE/near-MLSE (IDID$\geq$NT$-$1).

KADEL: the number of baud intervals between the finish and start of a training burst, for the PTS scheme of chapter 8. In this thesis KADEL=90.

DELT0: the threshold level H_{t} for the RTS scheme of chapter 8.

DAMSE: the weighting parameter λ for evaluating H_k (for the RTS scheme of chapter 8).

NDURE: the number of baud intervals of transmission simulated.

ITRIT: the warm-up period, in baud intervals.

ISED1, ISED2: starting seeds for the random variation of the data symbol and noise sample sequences respectively.

REVISE: logical variable, enabling operation of the revised SRK algorithm of [16].

TRAIN: logical variable, enabling operation in modes I and II when .TRUE., and mode III when .FALSE..

MLDON: logical variable, enabling operation of the MLSE/near-MLSE when .TRUE..

KALON: logical variable, used to switch between the SRK and SD algorithms.

PROBE: logical variable, used to turn ON/OFF the evaluation of the analytical P_e estimate.

The following variables are derived from the inputted variables, in the subroutine INITY.

SVAR, PIFF: the value of σ_s^2 and σ_s respectively.

NINQ, NSPA: the value of $L^2/4$ and $L/2$ respectively.

SIGM: the value of $\rho/2$.

QUE: the value of $(1/\omega)-1$, used in the SRK algorithm routine SRKALG.

IL, IL1: used in the routine MLSE for the operation of the MLSE/near-MLSE.

The remaining variables of significance are:

ITIME: the current number of baud intervals simulated during a program run.

IECNT(I): holds the error counts (both threshold detected and differential detected) for the MLSE/near-MLSE and DFE.

KINP(I): holds the detected symbols of the DFE, with detected symbols being used in the feedback process.

KINP1(I): holds the detected symbols of the DFE, with transmitted symbols being used in the feedback process.

IDECS(I): holds the detected symbols of the MLSE/near-MLSE.

INP(I): holds the transmitted symbols.

Note: KINP, KINP1, IDECS and INP store the data symbol as an integer in the range 0–(L^2-1).

ERRID: (complex) holds the error $(\tilde{s}_k-s_k)/\sigma_s$ of the DFE.

SQERR: holds the error $|\tilde{s}_k-s_k|^2$.

AVSQER: the time average of SQERR

SQISI: at any particular instant, holds the mean square ISI in detection of the DFE.

AVISI: the time average of SQISI.

SQNOI: at any particular instant, holds the mean square noise in detection of the DFE.

Note: SQISI+SQNOI is the mean square error in detection of the DFE.

AVNOI: the time average of SQNOI.

PEULB(I): holds the values of the probability of error as obtained from the analytical bounds derived in Appendix E.

SIRCH(I): holds the elements of the channel estimate.

SIRER: (complex) holds the algorithm error ϵ_k/σ_s

AVSIRE: the time average of $|\text{SIRER}|^2$.

SQCER: the error in the channel estimate, $\|\underline{y}_k-\underline{c}_{k-1}\|^2$.

AVCER: the time average of SQCER.

SHUK(K, I): (complex) holds the value of the I^{th} component of the channel response vector as a function of the time index K.

EST(I): (complex) the feedback taps of the DFE.

RALF(K): (complex) holds successive input samples r_k.

ALF(I): (complex) the feedforward taps of the DFE.

RVEC(I): (complex) successive outputs of the feedforward section of the DFE.

UUU(I): (complex) holds the elements of the upper triangular portion (excluding diagonal) of matrix $\mathbf{U}_k$ in the SRK algorithm.

DDD(I): holds the elements of diagonal matrix $\mathbf{D}_k$ in the SRK algorithm.

MAPX(K, I): holds the survivor sequences of the MLSE/near-MLSE.

COST(K): holds the costs of the survivor sequences in MAPX(K, I).

COPPS: a variable used for generating the Doppler shift effect in the channel response (not used in this thesis).

IREED(I): holds the seed for the random variation of the I^{th} path in the channel response.

RBV(I, K): (complex) holds the sequence $\{n_{mi}\}$ for producing the random variation in the I^{th} path of the channel response (see section 4.4.3).

GFV(I), DVFG(I): these define the approximate gaussian function in fig. 4.4.

IPETRA(I): hold the training symbol values for the PTS and RTS schemes of chapter 8.

IPETCO: used to increment the argument of IPETRA(I).

SAVSQE: the quantity H_k used in the RTS scheme of chapter 8.

TRANE, FREEZE: logical variables used in the PTS and RTS schemes.

KNOT, KNOT1: variables used in the PTS and RTS schemes. KNOT1 counts the number of baud intervals spent on training.

ICCP: used, in conjunction with IFADR, to determine when the sequence $\{n_{mi}\}$ is shifted along.

THOLD: logical variable that holds the value of TRAIN while the warm-up period is underway (ITIME<ITRIT) as TRAIN must always be .TRUE. during this period.

Subroutines and Functions

RINIT: this routine initialises IREED(.) and sets the values for GFV(.) and DVFG(.).

INITY: this routine initialises all other variables.

CHAN: this routine generates the current value of the fading channel response.

GEN: produces the data symbols, gaussian noise, received samples and outputs of the feedforward filter.

MLSE: implements the MLSE/near-MLSE detection algorithm.

QUANT: produces the detected symbols of the DFE.

EVAL: produces the analytical P_e estimate, and the mean square ISI and mean square noise values for the DFE.

SRKALG: implements the SRK algorithm.

SDALG: implements the SD algorithm.

DTAPS: computes the DFE taps from the channel estimate, as described in Appendix H.

IDECOD: produces the differential detected symbol (IX, IY) from two given symbol values IRE0 and IRE1 (which are in integer form, i.e. 0–(L^2-1)).

CGAUSS: (complex function) produces a complex gaussian noise sample of variance 2, as described in Appendix D.

CONSIG: (complex function) converts an integer in the range 0–(L^2-1) into a complex data symbol value, divided by σ_s.

MDECON: (function) converts a complex data symbol value into an integer in the range 0–(L^2-1).

RESULT: this routine (not detailed) is for outputting the results, which can be to a VDU screen, or to the file containing the input values, or to a separate file altogether. In our case the results were written in to the file containing the input values, and simulations were run during a batch processing session because of the length of time taken.

How the results are produced.

We will mention what routines are needed to produce the simulation results of this thesis. It may well be that not all of the variables will be required during a particular simulation run. When the fade rate is 1.0 Hz we set NDURE=64000 and ITRIT=4000, and when the fade rate is 2.0 Hz we set NDURE=32000 and ITRIT=2000.

The results for the DFE and MLSE/near-MLSE in modes II and III are produced from the program MAIN. For mode II the value of TRAIN is .TRUE., whereas for mode III it is .FALSE.. PEULB(1) and PEULB(2), evaluated in EVAL, are the lower and upper P_e bounds as described in Appendix E; PEULB(3) and PEULB(4) are the respective time averages of these bounds. The variables TRANE and FREEZE are only used in the PTS and RTS schemes, and are otherwise always .FALSE.. When the PTS and RTS schemes are used, the subroutine SRKALG must be modified slightly by the insertion of extra program lines, as is indicated in the subroutine listing. In the PTS scheme the value of KNOT, which is 10, specifies the length of the training burst. Furthermore, in the PTS scheme the logical variable FREEZE is not really needed.

When the DFE is implemented by the conventional SRK method, the subroutine DTAPS is not needed, and SRKALG is replaced by the subroutine SRKAL1, whose "call" statement in MAIN is CALL SRKAL1(RVEC(IDID+1),REVISE,TRAIN).

The results of chapter 6 (excluding the convergence curves of figs. 6.10 and 6.11) are also produced from the program MAIN, but the subroutines MLSE, QUANT, EVAL and DTAPS are not needed. The convergence curves of figs. 6.10, 6.11, 7.1–7.4 are produced by setting NDURE=81 and ITRIT=20, and including an outer DO loop in program MAIN just before the one going from 1 to NDURE, this outer loop going from 1 to 30. Also, the subroutine CONVER(VERG,ENCE) must be included in a "call" statement just before CALL DTAPS for the DFE-via-CE-method, and just after CALL GEN for the conventional SRK method. Subroutines MLSE and QUANT are not needed, and PROBE is .FALSE., and also the routines EVAL and DTAPS are not needed for figs. 6.10 and 6.11. The vectors VERG and ENCE should be appropriately dimensioned in MAIN and initialised to zero, and they hold, respectively, the successive (in time) values of the channel error and mean square error in detection. The "call" statement for subroutine CONVER is

```
IF (ITIME.GE.ITRIT) CALL CONVER(VERG,ENCE)
```

To produce the E.C. results for the MLSE/near-MLSE and DFE(6,*g*) in mode I, the program MAIN1 is used, which is identical to MAIN until just after the line "ITIME=0" in MAIN; the remaining lines in MAIN1 are

```
      PROGRAM MAIN1
          ⋮
      TRAIN=.TRUE.
      DO 1 J=1,NDURE
          ICCP=ICCP+1
          ICCP=MOD(ICCP,IFADR)
          CALL CHAN(ICCP,IFADR)
          CALL GEN
          CALL DTAPS1
          CALL MLSE(RVEC(IDID+1),RALF(IDID+1))
          CALL QUANT(RVEC(IDID+1),TRANE)
1     CONTINUE
      CALL RESULT
      STOP
      END
```

The subroutine DTAPS1 evaluates the DFE taps according to eqs. (2.137) and (2.133), using a complex version of the process given in [86] to solve (2.137). This is based on decomposing the positive definite Hermetian matrix $\mathbf{Y}_k^*$ as $\mathbf{Y}_k^*=\tilde{\mathbf{L}}\tilde{\mathbf{D}}\tilde{\mathbf{L}}^{*\mathbf{t}}$, where $\tilde{\mathbf{L}}$ is a lower triangular matrix with unit diagonal elements and $\tilde{\mathbf{D}}$ is a diagonal matrix with real positive elements. Vector $\underline{\mathbf{w}}_k$ is obtained by first solving $\underline{\mathbf{v}}=\tilde{\mathbf{L}}^{-1}\underline{\mathbf{y}}_{k0}^*$, the elements $\{v_i\}$ of $\underline{\mathbf{v}}$ being evaluated in the order $i=1$ to N. Then the equation $\underline{\mathbf{w}}_k=(\tilde{\mathbf{L}}^{*\mathbf{t}})^{-1}\tilde{\mathbf{D}}^{-1}\underline{\mathbf{v}}$ is solved, with the elements $\{w_{ki}\}$ of $\underline{\mathbf{w}}_k$ being evaluated in the order $i=N-1$ to 0.

To produce the results for the DFE(N,g)-I_1, simultaneously for $L^2=4$, 16 and 64, the routines GEN, MLSE and QUANT are not required in MAIN1, and the routine EVAL1 should be included in the statement CALL EVAL1 immediately after CALL DTAPS1. The two quantities SQISI and SQNOI, evaluated in EVAL1, respectively denote the mean square ISI and mean square noise, divided by σ_s^2. PEULB(1)–PEULB(4) contain the analytical P_e values for $L^2=4$, while PEULB(5)–PEULB(8) and PEULB(9)–PEULB(12) are for $L^2=16$ and 64 respectively. In our case we set IFADR to 108, with NDURE=32000 and ITIME=2000.

The results for the ZF MFE and MMSE MFE of chapter 5 are produced from MAIN1, but with routines GEN, DTAPS1, MLSE and QUANT absent. The routine EVAL2 should be included in the "call" statement CALL EVAL2 immediately after CALL CHAN(ICCP,IFADR). SQERR and SQISI, evaluated in EVAL2, denote the mean square error of the ZF MFE and MMSE MFE respectively. PEULB(1), PEULB(3) and PEULB(5) contain the instantaneous error probability values for the ZF MFE for L^2=4, 16 and 64 respectively, with PEULB(2), PEULB(4) and PEULB(6) being the respective time averages. In our case we set IFADR to 108, with NDURE=32000 and ITIME=2000.

```
      PROGRAM MAIN
      COMPLEX ERRID,SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,UUU,SPEC,RBV
      LOGICAL TRAIN,KALON,MLDON,REVISE,THOLD,TRANE,FREEZE,PROBE
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
      COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
      COMMON/KALSRK/UUU(410),DDD(30),EPSIL,QUE,HQUEP
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      COMMON/MLD/LEMM,MUE,MAPX(16,20),COST(16)
      COMMON/VARI/SPEC(10),STAND(10),DOPPS,COPPS
      COMMON/RAYL/IREED(10),RBV(10,13),GFV(6),DVFG(6)
      COMMON/CONSIY/NINQ,NSPA
      COMMON/PETRAN/IPETCO,IPETRA(3),SAVSQE,TRANE,FREEZE,KNOT,KNOT1
      CALL VALUES(NDURE,IFADR,TRAIN,KALON,MLDON,REVISE,PROBE)
      CALL RINIT
      ICCP=-1
      CALL INITY(ICCP,IFADR)
      ITIME=0
      THOLD=TRAIN
      TRAIN=.TRUE.
      DO 1 J=1,NDURE
         ICCP=ICCP+1
         ICCP=MOD(ICCP,IFADR)
         CALL CHAN(ICCP,IFADR)
         CALL GEN
         IF (MLDON) CALL MLSE(RVEC(IDID+1),RALF(IDID+1))
         CALL QUANT(RVEC(IDID+1),TRANE)
         CALL EVAL(PROBE)
         IF (KALON) THEN
         CALL SRKALG(RVEC(IDID+1),RALF(IDID+1),REVISE,TRAIN)
         ELSE
         CALL SDALG(RVEC(IDID+1),RALF(IDID+1),TRAIN)
         ENDIF
         CALL DTAPS
         IF (ITIME.EQ.ITRIT) TRAIN=THOLD
 1    CONTINUE
```

```
      CALL RESULT
      STOP
      END


      SUBROUTINE RINIT
      COMPLEX RBV,CGAUSS
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/RAYL/IREED(10),RBV(10,13),GFV(6),DVFG(6)
      IREED(1)=864095000
      IREED(2)=468381000
      IREED(3)=946715000
      IREED(4)=622081000
      IREED(5)=713574000
      IREED(6)=579432000
      IREED(7)=105242000
      IREED(8)=333333000
      IREED(9)=212525000
      IREED(10)=55533000
      DO 1 I=1,NT
         DO 2 J=2,13
            RBV(I,J)=CGAUSS(IREED(I))
2        CONTINUE
1     CONTINUE
      TPI=2.5*SQRT(3.141592654)
      TPI=1./SQRT(TPI)
      X=2.2
      DO 3 I=1,6
         GFV(I)=TPI*EXP(-X*X/2.)
         X=X-0.4
3     CONTINUE
      DO 4 I=1,5
         DVFG(I)=GFV(I+1)-GFV(I)
4     CONTINUE
      DVFG(6)=TPI-GFV(6)
      RETURN
      END
```

```
 SUBROUTINE INITY(ICCP,IFADR)
 COMPLEX ERRID,SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,UUU,SPEC,
+CONSIG,ROCK
 LOGICAL TRANE,FREEZE
 COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
 COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
 COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
 COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
 COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
 COMMON/KALSRK/UUU(410),DDD(30),EPSIL,QUE,HQUEP
 COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
 COMMON/MLD/LEMM,MUE,MAPX(16,20),COST(16)
 COMMON/VARI/SPEC(10),STAND(10),DOPPS,COPPS
 COMMON/CONSIY/NINQ,NSPA
 COMMON/PETRAN/IPETCO,IPETRA(3),SAVSQE,TRANE,FREEZE,KNOT,KNOT1
 SVAR=2.*FLOAT(NSL*NSL-1)/3.
 PIFF=SQRT(SVAR)
 NSPA=NSL/2
 NINQ=NSL*NSL/4
 SIGM=0.5/(10.**(SNR0/10.))
 QUE=HQUEP-1.
 COPPS=0.
 IF (NSL.EQ.2) THEN
 IL=NSL**(2*(NT-1))
 IL1=NSL**(2*(NT-2))
 ELSE
 IL=LEMM*MUE*MUE
 IL1=MUE*MUE
 ENDIF
 J=0
 IF (NSL.EQ.4) THEN
 J=3
 ELSE IF (NSL.EQ.8) THEN
 J=10
 ENDIF
 IPETRA(1)=J
 IPETRA(2)=J+(NSL*NSL)/4
 IPETRA(3)=J+(3*NSL*NSL)/4
```

```
      IPETCO=0
      DO 6 I=1,(IDID+LANT+NT-1)
         INP(I)=IPETRA(IPETCO+1)
         IPETCO=MOD((IPETCO+1),NT)
6     CONTINUE
      LUN=LEMM
      IF (NSL.EQ.2) LUN=IL
      DO 1 I=1,LUN
         J=I-1
         DO 2 K=1,(NT-1)
            MAPX(I,K)=MOD(J,(NSL*NSL))
            J=J/(NSL*NSL)
2        CONTINUE
         DO 3 M=NT,IDID
            MAPX(I,M)=0
3        CONTINUE
         COST(I)=1.E3
1     CONTINUE
      DO 10 I=1,LANT
         ALF(I)=(0.,0.)
         DO 17 J=1,NT
            SHUK(I,J)=(0.,0.)
17       CONTINUE
10    CONTINUE
      ITIME=0
      DO 11 I=1,(IDID+LANT)
         ICCP=MOD((ICCP+1),IFADR)
         CALL CHAN(ICCP,IFADR)
         ROCK=(0.,0.)
         DO 5 J=1,NT
            ROCK=ROCK+CONSIG(INP(I+NT-J))*SHUK(LANT,J)
5        CONTINUE
         RALF(I)=ROCK
11    CONTINUE
      DO 7 I=1,(IDID+1)
         RVEC(I)=(0.,0.)
         KINP(I)=INP(I+NT-1)
         KINP1(I)=KINP(I)
```

```
7    CONTINUE
     DO 12 I=1,NT
         IDECS(I)=INP(I)
12   CONTINUE
     J=(LANT+NT-1)*(LANT+NT-2)
     DO 13 I=1,(J/2)
          UUU(I)=(0.,0.)
13   CONTINUE
     DO 14 I=1,(LANT+NT-1)
          DDD(I)=1.
14   CONTINUE
     DO 15 I=1,5
          IECNT(I)=0
15   CONTINUE
     DO 8 I=1,12
     PEULB(I)=0.
8    CONTINUE
     EST(1)=(1.,0.)
     SIRCH(1)=(0.,0.)
     DO 9 I=2,NT
         EST(I)=(0.,0.)
         SIRCH(I)=(0.,0.)
9    CONTINUE
     ERRID=(0.,0.)
     SQERR=0.
     AVSQER=0.
     SQISI=0.
     AVISI=0.
     SQNOI=0.
     AVNOI=0.
     SIRER=(0.,0.)
     AVSIRE=0.
     SQCER=0.
     AVCER=0.
     SAVSQE=0.
     IPETCO=0
     KNOT=0
     KNOT1=0
```

```
      TRANE=.FALSE.
      FREEZE=.FALSE.
      RETURN
      END


      SUBROUTINE CHAN(ICCP,IFADR)
      COMPLEX SIRCH,SIRER,SHUK,EST,SPEC,RBV,CGAUSS,SUMN
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      COMMON/VARI/SPEC(10),STAND(10),DOPPS,COPPS
      COMMON/RAYL/IREED(10),RBV(10,13),GFV(6),DVFG(6)
      ITIME=ITIME+1
      DO 7 I=1,(LANT-1)
          DO 8 J=1,NT
              SHUK(I,J)=SHUK(I+1,J)
 8        CONTINUE
 7    CONTINUE
      COPPS=AMOD((COPPS+DOPPS),6.283185308)
      IF (ICCP.NE.0) GOTO 3
      DO 1 I=1,NT
          DO 2 J=1,12
              RBV(I,J)=RBV(I,J+1)
 2        CONTINUE
          RBV(I,13)=CGAUSS(IREED(I))
 1    CONTINUE
 3    PATL=FLOAT(ICCP)/FLOAT(IFADR)
      DO 4 I=1,NT
          SUMN=(0.,0.)
          DO 5 J=1,5
      SUMN=SUMN+(GFV(J)+PATL*DVFG(J))*RBV(I,13-J)
 5        CONTINUE
          DO 6 J=6,2,-1
      SUMN=SUMN+(GFV(J)-PATL*DVFG(J-1))*RBV(I,J)
 6        CONTINUE
          SUMN=SUMN+PATL*GFV(1)*RBV(I,13)+
     +     (1.-PATL)*GFV(1)*RBV(I,1)
```

```
         X=PATL
         IF (PATL.GT.0.5) X=1.-PATL
      SUMN=SUMN+(GFV(6)+2.*X*DVFG(6))*RBV(I,7)
      SHUK(LANT,I)=(SPEC(I)+SUMN*STAND(I))*CMPLX(COS(COPPS),SIN(COPPS))
4     CONTINUE
      RETURN
      END


      SUBROUTINE GEN
      COMPLEX SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,CGAUSS,CONSIG,RECD
      LOGICAL TRANE,FREEZE
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      COMMON/PETRAN/IPETCO,IPETRA(3),SAVSQE,TRANE,FREEZE,KNOT,KNOT1
      CALL RANSET(ISED1)
7     UNIF=RANF()
      IF ((UNIF.LE.0.).OR.(UNIF.GE.1.)) GOTO 7
      IX=IFIX(FLOAT(NSL*NSL)*UNIF)
      CALL RANGET(ISED1)
      DO 1 I=1,(IDID+LANT+NT-2)
         INP(I)=INP(I+1)
1     CONTINUE
      IF (FREEZE) THEN
      IX=IPETRA(IPETCO+1)
      IPETCO=MOD((IPETCO+1),NT)
      ENDIF
      INP(IDID+LANT+NT-1)=IX
      RECD=(0.,0.)
      DO 2 I=1,NT
         RECD=RECD+SHUK(LANT,I)*CONSIG(INP(IDID+LANT+NT-I))
2     CONTINUE
      RECD=RECD+SQRT(SIGM)*CGAUSS(ISED2)
      DO 3 I=1,(IDID+LANT-1)
          RALF(I)=RALF(I+1)
```

```
3    CONTINUE
     RALF(IDID+LANT)=RECD
     RECD=(0.,0.)
     DO 5 I=1,LANT
         RECD=RECD+ALF(I)*RALF(IDID+I)
5    CONTINUE
     DO 6 I=1,IDID
         RVEC(I)=RVEC(I+1)
6    CONTINUE
     RVEC(IDID+1)=RECD
     RETURN
     END


     SUBROUTINE MLSE(RECD,RECD1)
     COMPLEX RECD,RECD1,SIRCH,SIRER,SHUK,EST,CONSIG,DEKAY
     DIMENSION IMAPX(16,20),IMAPX0(64),IMAPX1(64),COSVA0(64),COSVA1(64)
     COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
     COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
     COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
     COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
     COMMON/MLD/LEMM,MUE,MAPX(16,20),COST(16)
     IF (ITIME.LT.ITRIT) RETURN
     IF (ITIME.EQ.ITRIT) THEN
     J=1
     IF (NSL.EQ.2) THEN
     DO 3 K=1,(NT-1)
         J=J+INP(IDID+NT+1-K)*(NSL**(2*(K-1)))
3    CONTINUE
     ELSE
     DO 4 K=1,(NT-1)
         MAPX(1,K)=INP(IDID+NT+1-K)
4    CONTINUE
     ENDIF
     DO 5 K=NT,IDID
         MAPX(J,K)=INP(IDID+NT+1-K)
5    CONTINUE
     IDECS(NT)=INP(NT)
```

```
      COST(J)=0.
      RETURN
      ENDIF
      TLIM=1.E30
      IF (NSL.EQ.2) GOTO 38
C     With reference to the near-MLSE algorithm of Appendix G, IL and IL1 are equivalent to
C     $mn^2$ and $n^2$ respectively, and DEKAYR and DEKAYI are equivalent to Re[$\nu_k$] and Im[$\nu_k$]
C     respectively.
      DO 1 I=1,LEMM
          IQ=(I-1)*IL1
          DEKAY=RECD
          DO 2 J=2,NT
              LLL=MAPX(I,J-1)
              DEKAY=DEKAY-EST(J)*CONSIG(LLL)
2         CONTINUE
          DEKAYR=PIFF*REAL(DEKAY)
          DEKAYI=PIFF*AIMAG(DEKAY)
          M1=IQ+MUE
          M2=M1+MUE
          CPRR=ABS(DEKAYR)
          J=IFIX(CPRR)
C     Select first sequence from Re[$\nu_k$].
          IF (J.GT.(NSL-1)) THEN
          J=NSL-1
          ELSE
          J=J+MOD((J+1),2)
          ENDIF
C     Select second sequence from Re[$\nu_k$].
          CPRR=CPRR-FLOAT(J)
          IF (CPRR.GE.0.) THEN
          K=J+2
          ELSE
          K=J-2
          ENDIF
          IF (K.GT.(NSL-1)) K=K-4
          IF (DEKAYR.LT.0.) THEN
          J=-J
          K=-K
      ENDIF
      COSVA0(IQ+1)=ABS(CPRR)
      COSVA0(IQ+2)=ABS(DEKAYR−FLOAT(K))
      IMAPX0(IQ+1)=J
      IMAPX0(IQ+2)=K
      CPRI=ABS(DEKAYI)
      J=IFIX(CPRI)
      IF (J.GT.(NSL−1)) THEN
      J=NSL−1
      ELSE
      J=J+MOD((J+1),2)
      ENDIF
      CPRI=CPRI−FLOAT(J)
      IF (CPRI.GE.0.) THEN
      K=J+2
      ELSE
      K=J−2
      ENDIF
      IF (K.GT.(NSL−1)) K=K−4
      IF (DEKAYI.LT.0.) THEN
      J=−J
      K=−K
      ENDIF
      COSVA0(M1+1)=ABS(CPRI)
      COSVA0(M1+2)=ABS(DEKAYI−FLOAT(K))
      IMAPX0(M1+1)=J
      IMAPX0(M1+2)=K
C   Evaluate the costs of the sequences and find the one with the lowest.
      DO 11 J=1,MUE
         DO 12 K=1,MUE
      CPR=COST(I)+COSVA0(IQ+J)**2+COSVA0(M1+K)**2
      LLL=IQ+(J−1)*MUE+K
      COSVA1(LLL)=CPR
      IMAPX1(LLL)=MDECON(IMAPX0(IQ+J),IMAPX0(M1+K))
      IF (CPR.GT.TLIM) GOTO 12
      TLIM=CPR
      IACF=LLL
12       CONTINUE
```

```
11          CONTINUE
1     CONTINUE
C     Address of lowest-cost sequence, and detected symbol.
      IDENT=(IACF-1)/IL1+1
      LIMT=MAPX(IDENT,IDID)
      DO 13 I=1,(NT-1)
          IDECS(I)=IDECS(I+1)
13    CONTINUE
      IDECS(NT)=LIMT
      DO 14 I=2,IDID
          IMAPX(1,I)=MAPX(IDENT,I-1)
14    CONTINUE
      IMAPX(1,1)=IMAPX1(IACF)
C     Lowesr-cost sequence effectively removed from rest by putting high cost value.
      COSVA1(IACF)=1.E60
C     Sequences for which $\alpha_{k-q} \neq \hat{s}_{k-q}$ are removed.
      DO 15 I=1,LEMM
          IQ=(I-1)*IL1
          IF (MAPX(I,IDID).EQ.LIMT) GOTO 15
          DO 16 J=1,IL1
              COSVA1(IQ+J)=1.E60
16          CONTINUE
15    CONTINUE
      IREST=1
C     Any sequences originating from lowest-cost survivor? If so, find the lowest-cost one, store
C     it and its cost, and then effectively remove it from the rest.
      IF (MAPX(1,IDID).EQ.LIMT) THEN
      TLIM0=1.E30
      DO 17 I=1,IL1
          CPR=COSVA1(I)
          IF (CPR.GT.TLIM0) GOTO 17
          TLIM0=CPR
          IACF=I
17    CONTINUE
      COST(2)=TLIM0-TLIM
      DO 18 I=2,IDID
          IMAPX(2,I)=MAPX(1,I-1)
18    CONTINUE
```

```
      IMAPX(2,1)=IMAPX1(IACF)
      COSVA1(IACF)=1.E60
      IREST=2
      ENDIF
C     Select rest of sequences.
      DO 19 I=(IREST+1),LEMM
         CPR=1.E30
         DO 20 J=1,IL
            IF (COSVA1(J).GE.CPR) GOTO 20
            CPR=COSVA1(J)
            IACF=J
20       CONTINUE
         COST(I)=CPR
         COSVA1(IACF)=1.E60
         IMAPX0(I)=IACF
19    CONTINUE
C     Sort sequences for start of next sampling instant.
      DO 24 I=(IREST+1),LEMM
         IDENT=(IMAPX0(I)-1)/IL1+1
         DO 25 J=2,IDID
            IMAPX(I,J)=MAPX(IDENT,J-1)
25       CONTINUE
         IMAPX(I,1)=IMAPX1(IMAPX0(I))
24    CONTINUE
      DO 22 I=1,LEMM
         DO 23 J=1,IDID
            MAPX(I,J)=IMAPX(I,J)
23       CONTINUE
22    CONTINUE
      RETURN
C     Start of MLSE for 4 QAM. IL and IL1 are respectively equal to $L^{2g}$ and $L^{2(g-1)}$.
38    DO 30 I=0,(IL1-1)
         DO 31 K=0,(NSL*NSL-1)
            JK=I-IL1+1
            CT=1.E30
            DO 32 J=0,(NSL*NSL-1)
               JK=JK+IL1
               DEKAY=RECD1-CONSIG(K)*SIRCH(1)
```

```
                  DO 33 M=1,(NT-1)
      LLL=MAPX(JK,M)
      DEKAY=DEKAY-CONSIG(LLL)*SIRCH(M+1)
33                CONTINUE
                  CPR=COST(JK)+CABS(DEKAY)**2
                  IF (CPR.GT.CT) GOTO 32
                  CT=CPR
                  IACF=JK
32              CONTINUE
              KADD=I*NSL*NSL+K+1
              COSVA0(KADD)=CT
              DO 34 L=(NT-1),(IDID-1)
                  IMAPX(KADD,L)=MAPX(IACF,L)
34              CONTINUE
              IF (TLIM.LE.CT) GOTO 31
              TLIM=CT
              LIMT=MAPX(IACF,IDID)
31          CONTINUE
30    CONTINUE
      DO 35 I=1,IL
          DO 36 L=NT,IDID
              MAPX(I,L)=IMAPX(I,L-1)
36          CONTINUE
          COST(I)=COSVA0(I)-TLIM
35    CONTINUE
      DO 37 I=1,(NT-1)
          IDECS(I)=IDECS(I+1)
37    CONTINUE
      IDECS(NT)=LIMT
      RETURN
      END


      SUBROUTINE QUANT(ARR,TRANE)
      COMPLEX ARR,SIRCH,SIRER,SHUK,EST,CONSIG,RISI,XY
      LOGICAL TRANE
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      KINP(22)=KINP(1)
      DO 1 I=1,IDID
         KINP(I)=KINP(I+1)
         KINP1(I)=KINP1(I+1)
1     CONTINUE
      IF ((ITIME.LE.ITRIT).OR.TRANE) THEN
      KINP(IDID+1)=INP(IDID+NT)
      KINP1(IDID+1)=INP(IDID+NT)
      RETURN
      ENDIF
      RISI=(0.,0.)
      DO 2 I=2,NT
         RISI=RISI+CONSIG(KINP(IDID+2-I))*EST(I)
2     CONTINUE
      RISI=(ARR-RISI)*PIFF
      RX=REAL(RISI)
      RY=AIMAG(RISI)
      IX=IFIX(ABS(RX))
      IY=IFIX(ABS(RY))
      IF (MOD(IX,2).EQ.0) IX=IX+1
      IF (MOD(IY,2).EQ.0) IY=IY+1
      I=NSL-1
      IF (IX.GT.I) IX=I
      IF (IY.GT.I) IY=I
      IF (SIGN(1.,RX).LT.0.) IX=-IX
      IF (SIGN(1.,RY).LT.0.) IY=-IY
      KINP(IDID+1)=MDECON(IX,IY)
      RISI=(0.,0.)
      DO 5 I=2,NT
         RISI=RISI+CONSIG(INP(IDID+NT+1-I))*EST(I)
5     CONTINUE
      RISI=(ARR-RISI)*PIFF
      RX=REAL(RISI)
      RY=AIMAG(RISI)
      IX=IFIX(ABS(RX))
      IY=IFIX(ABS(RY))
      IF (MOD(IX,2).EQ.0) IX=IX+1
      IF (MOD(IY,2).EQ.0) IY=IY+1
      I=NSL-1
      IF (IX.GT.I) IX=I
      IF (IY.GT.I) IY=I
      IF (SIGN(1.,RX).LT.0.) IX=-IX
      IF (SIGN(1.,RY).LT.0.) IY=-IY
      KINP1(IDID+1)=MDECON(IX,IY)
      CALL IDECOD(INP(NT),INP(NT-1),IIX,IIY)
      CALL IDECOD(IDECS(NT),IDECS(NT-1),IDX,IDY)
      I=IABS(IIX-IDX)+IABS(IIY-IDY)
      IF (I.EQ.0) GOTO 8
      IECNT(1)=IECNT(1)+1
 8    IF (IDECS(NT).NE.INP(NT)) IECNT(2)=IECNT(2)+1
      CALL IDECOD(KINP(1),KINP(22),IDX,IDY)
      I=IABS(IIX-IDX)+IABS(IIY-IDY)
      IF (I.EQ.0) GOTO 9
      IECNT(3)=IECNT(3)+1
 9    IF (KINP(1).NE.INP(NT)) IECNT(4)=IECNT(4)+1
      IF (KINP1(1).NE.INP(NT)) IECNT(5)=IECNT(5)+1
      RETURN
      END


      SUBROUTINE EVAL(PROBE)
      COMPLEX ERRID,SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,CONSIG,XY
      DIMENSION CLASS(180)
      LOGICAL TRANE,FREEZE,PROBE
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      COMMON/PETRAN/IPETCO,IPETRA(3),SAVSQE,TRANE,FREEZE,KNOT,KNOT1
      IF (ITIME.LE.ITRIT) RETURN
      IF (TRANE.OR.FREEZE) RETURN
      IIX=IFIX(0.5+LOG10(FLOAT(NSL))/LOG10(2.))
      IIY=IIX*(2*(LANT+NT-1)-1)
```

```
C     Evaluate the ISI terms.
      XX=0.
      B1=0.
      MM=0
      DO 2 I=1,(LANT−1)
          XY=(0.,0.)
          DO 3 J=(I+1),MIN0(LANT,(NT+I))
              XY=XY+ALF(J)*SHUK(J,J−I)
3         CONTINUE
          C1=ABS(REAL(XY))
          C2=ABS(AIMAG(XY))
          DO 18 J=1,IIX
              B2=FLOAT(NSL/(2**J))
              CLASS(MM+1)=C1*B2
              CLASS(MM+2)=C2*B2
              MM=MM+2
18        CONTINUE
          XX=XX+CABS(XY)**2
          B1=B1+CABS(ALF(I))**2
2     CONTINUE
      SQISI=XX
      B1=B1+CABS(ALF(LANT))**2
      SQNOI=2.*SIGM*SVAR*B1
      XY=(0.,0.)
      DO 4 I=1,MIN0(LANT,NT)
          XY=XY+ALF(I)*SHUK(I,I)
4     CONTINUE
      XY=XY−1.
      A1=REAL(XY)
      C2=ABS(AIMAG(XY))
      DO 19 J=1,IIX
          B2=FLOAT(NSL/(2**J))
          CLASS(MM+1)=C2*B2
          MM=MM+1
19    CONTINUE
      SQISI=SQISI+CABS(XY)**2
      XX=0.
      DO 5 I=1,(NT−1)
```

```
          XY=(0.,0.)
          DO 6 J=1,MIN0(LANT,(NT-I))
              XY=XY+ALF(J)*SHUK(J,J+I)
6         CONTINUE
          XY=XY-EST(I+1)
          C1=ABS(REAL(XY))
          C2=ABS(AIMAG(XY))
          DO 20 J=1,IIX
              B2=FLOAT(NSL/(2**J))
              CLASS(MM+1)=C1*B2
              CLASS(MM+2)=C2*B2
              MM=MM+2
20        CONTINUE
          XX=XX+CABS(XY)**2
5     CONTINUE
      SQISI=SVAR*(SQISI+XX)
      PERRI=FLOAT(ITIME-ITRIT-KNOT1)
      AVISI=AVISI+(SQISI-AVISI)/PERRI
      AVNOI=AVNOI+(SQNOI-AVNOI)/PERRI
      IF (PROBE) RETURN
C     Sort the ISI terms, and evaluate the upper (F2) and lower (F1) error bounds according to
C     the method given in Appendix E.
      DO 21 I=1,(IIY-1)
          B1=0.
          DO 22 J=I,IIY
              IF (B1.GT.CLASS(J)) GOTO 22
              B1=CLASS(J)
              MM=J
22        CONTINUE
          IF (B1.EQ.0.) THEN
          IIY=I-1
          GOTO 23
          ENDIF
          CLASS(MM)=CLASS(I)
          CLASS(I)=B1
21    CONTINUE
      IF (CLASS(IIY).EQ.0.) IIY=IIY-1
23    C1=1./SQRT(SQNOI)
```

```
      F1=0.
      F2=0.
      DO 12 I=2,(NSL/2)
         K=2*I-NSL-1
         D1=1.-FLOAT(K)*A1
         D2=1.+FLOAT(K)*A1
         E1=0.
         E2=0.
         B2=D1*C1
         IF (B2.LE.25.) E1=ERFC(B2)
         B2=D2*C1
         IF (B2.LE.25.) E2=ERFC(B2)
         IF (IIY.EQ.0) THEN
         F2=F2+E1+E2
         GOTO 12
         ENDIF
         C2=0.5
         F2=F2+C2*(E1+E2)
         DO 13 J=1,(IIY-1)
            D1=D1-CLASS(J)
            D2=D2-CLASS(J)
            E1=0.
            E2=0.
            B2=D1*C1
            IF (B2.LE.25.) E1=ERFC(B2)
            B2=D2*C1
            IF (B2.LE.25.) E2=ERFC(B2)
            C2=C2/2.
            F2=F2+C2*(E1+E2)
13          CONTINUE
         D1=D1-CLASS(IIY)
         D2=D2-CLASS(IIY)
         E1=0.
         E2=0.
         B2=D1*C1
         IF (B2.LE.25.) E1=ERFC(B2)
         B2=D2*C1
         IF (B2.LE.25.) E2=ERFC(B2)
```

```
          F2=F2+C2*(E1+E2)
          D1=2.*(1.-FLOAT(K)*A1)-D1
          D2=2.*(1.+FLOAT(K)*A1)-D2
          E1=0.
          E2=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          B2=D2*C1
          IF (B2.LE.25.) E2=ERFC(B2)
          F1=F1+E1+E2
12    CONTINUE
      D1=1.+FLOAT(NSL-1)*A1
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      IF (IIY.EQ.0) THEN
      F2=F2+E1
      GOTO 24
      ENDIF
      C2=0.5
      F2=F2+C2*E1
      DO 14 J=1,(IIY-1)
          D1=D1-CLASS(J)
          E1=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          C2=C2/2.
          F2=F2+C2*E1
14    CONTINUE
      D1=D1-CLASS(IIY)
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      F2=F2+C2*E1
      D1=2.*(1.+FLOAT(NSL-1)*A1)-D1
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      F1=F1+E1+F2
24     PEULB(1)=F1/FLOAT(NSL)
      PEULB(2)=2.*F2/FLOAT(NSL)
      PEULB(3)=PEULB(3)+(PEULB(1)-PEULB(3))/PERRI
      PEULB(4)=PEULB(4)+(PEULB(2)-PEULB(4))/PERRI
      RETURN
      END


      SUBROUTINE SRKALG(AR1,AR2,REVISE,TRAIN)
      COMPLEX AR1,AR2,ERRID,SIRCH,SIRER,SHUK,EST,UUU,RDASH,
     +CONSIG,FFF(30),GGG(30),HLAMDA,BETA1,XY
      DIMENSION AAA(30)
      LOGICAL REVISE,TRAIN,TRANE,FREEZE
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
      COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/KALSRK/UUU(410),DDD(30),EPSIL,QUE,HQUEP
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      COMMON/PETRAN/IPETCO,IPETRA(3),SAVSQE,TRANE,FREEZE,KNOT,KNOT1
      SIRER=(0.,0.)
      RDASH=(0.,0.)
      IF (.NOT.TRAIN) THEN
      DO 11 I=1,NT
          XY=CONSIG(KINP(IDID+2-I))
          SIRER=SIRER+XY*SIRCH(I)
          RDASH=RDASH+XY*EST(I)
11     CONTINUE
      ELSE
      DO 12 I=1,NT
          XY=CONSIG(INP(IDID+NT+1-I))
          SIRER=SIRER+XY*SIRCH(I)
          RDASH=RDASH+XY*EST(I)
12     CONTINUE
      ENDIF
      ERRID=AR1-RDASH
      SIRER=AR2-SIRER
      SQERR=SVAR*(CABS(ERRID)**2)
      SAVSQE=DAMSE*SQERR+(1.-DAMSE)*SAVSQE
      SQCER=0.
      DO 3 I=1,NT
          SQCER=SQCER+CABS(SIRCH(I)-SHUK(1,I))**2
3     CONTINUE
      IF (ITIME.LE.ITRIT) GOTO 4
```

INSERT point for extra lines for PTS and RTS schemes (see end of this routine)

```
      PERRI=FLOAT(ITIME-ITRIT)
      AVSQER=AVSQER+(SQERR-AVSQER)/PERRI
      AVSIRE=AVSIRE+(CABS(SIRER)**2-AVSIRE)/PERRI
      AVCER=AVCER+(SQCER-AVCER)/PERRI
4     IF (.NOT.TRAIN) THEN
      XY=CONSIG(KINP(IDID+1))
      ELSE
      XY=CONSIG(INP(IDID+NT))
      ENDIF
      FFF(1)=CONJG(XY)
      GGG(1)=DDD(1)*FFF(1)
      XY=GGG(1)*CONJG(FFF(1))
      AAA(1)=EPSIL+REAL(XY)
      MM=0
      DO 1 J=2,NT
          IF (.NOT.TRAIN) THEN
          XY=CONSIG(KINP(IDID+2-J))
          ELSE
          XY=CONSIG(INP(IDID+NT+1-J))
          ENDIF
          FFF(J)=CONJG(XY)
          DO 2 I=1,(J-1)
              MM=MM+1
              IF (.NOT.TRAIN) THEN
              XY=CONSIG(KINP(IDID+2-I))
              ELSE
              XY=CONSIG(INP(IDID+NT+1-I))
              ENDIF
              FFF(J)=FFF(J)+UUU(MM)*CONJG(XY)
2          CONTINUE
           GGG(J)=DDD(J)*FFF(J)
           XY=GGG(J)*CONJG(FFF(J))
           AAA(J)=AAA(J-1)+REAL(XY)
1     CONTINUE
      IF (REVISE) THEN
      HHTT=QUE*AAA(NT)
      ELSE
      HHTT=0.
      ENDIF
      CAPPA=1./(AAA(1)+HHTT)
      DDD(1)=DDD(1)*HQUEP*(EPSIL+HHTT)*CAPPA
      MM=0
      DO 6 J=2,NT
           BETA=AAA(J-1)+HHTT
           HLAMDA=FFF(J)*CAPPA
           CAPPA=1./(AAA(J)+HHTT)
           DDD(J)=DDD(J)*HQUEP*BETA*CAPPA
           DO 7 I=1,(J-1)
                MM=MM+1
                BETA1=UUU(MM)
                UUU(MM)=BETA1-CONJG(GGG(I))*HLAMDA
                GGG(I)=GGG(I)+GGG(J)*CONJG(BETA1)
7          CONTINUE
6     CONTINUE
      BETA1=SIRER/AAA(NT)
      DO 8 J=1,NT
           SIRCH(J)=SIRCH(J)+GGG(J)*BETA1
8     CONTINUE
      RETURN
      END

C     INSERT lines for PTS scheme.
      IF (FREEZE) THEN
      KNOT1=KNOT1+1
      IF (KNOT.GT.0) KNOT=KNOT-1
      IF (KNOT.EQ.0) THEN
      TRANE=.FALSE.
      FREEZE=.FALSE.
      ENDIF
      ELSE IF ((.NOT.TRAIN).AND.(.NOT.TRANE)) THEN
      IF (MOD(ITIME,(KADEL+10)).EQ.0) THEN
      FREEZE=.TRUE.
      TRANE=.TRUE.
      KNOT=10
      SIRCH(1)=(0.,0.)
      DO 5 I=2,NT
          EST(I)=(0.,0.)
          SIRCH(I)=(0.,0.)
5      CONTINUE
      J=(LANT+NT-1)*(LANT+NT-2)
      DO 9 I=1,J/2
          UUU(I)=(0.,0.)
9      CONTINUE
      DO 10 I=1,(LANT+NT-1)
          DDD(I)=1.
10      CONTINUE
      ENDIF
      ENDIF
C     End of INSERT lines for PTS scheme.

C     INSERT lines for RTS scheme.
      IF (FREEZE) THEN
      KNOT1=KNOT1+1
      IF (KNOT.GT.0) KNOT=KNOT-1
      IF (KNOT.EQ.1) TRANE=.TRUE.
      IF (KNOT.EQ.0) THEN
      IF (SAVSQE.LT.DELT0) THEN
      FREEZE=.FALSE.
      KNOT=LANT
      IF (KNOT.EQ.1) TRANE=.FALSE.
      ENDIF
      ENDIF
      ELSE IF ((.NOT.TRAIN).AND.(.NOT.TRANE)) THEN
      IF (SAVSQE.GE.DELT0) THEN
      FREEZE=.TRUE.
      KNOT=LANT
      IF (KNOT.EQ.1) TRANE=.TRUE.
      ENDIF
      ELSE IF (KNOT.GT.1) THEN
      KNOT1=KNOT1+1
      KNOT=KNOT-1
      IF (KNOT.EQ.1) TRANE=.FALSE.
      ENDIF
C     End of INSERT lines for RTS scheme.


      SUBROUTINE SDALG(AR1,AR2,TRAIN)
      COMPLEX AR1,AR2,ERRID,SIRCH,SIRER,SHUK,EST,RDASH,CONSIG,XY
      LOGICAL TRAIN
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
      COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      SIRER=(0.,0.)
      RDASH=(0.,0.)
      IF (.NOT.TRAIN) THEN
      DO 11 I=1,NT
         XY=CONSIG(KINP(IDID+2-I))
         SIRER=SIRER+XY*SIRCH(I)
         RDASH=RDASH+XY*EST(I)
11     CONTINUE
      ELSE
      DO 12 I=1,NT
         XY=CONSIG(INP(IDID+NT+1-I))
         SIRER=SIRER+XY*SIRCH(I)
         RDASH=RDASH+XY*EST(I)
12     CONTINUE
      ENDIF
      ERRID=AR1-RDASH
      SIRER=AR2-SIRER
      SQERR=SVAR*(CABS(ERRID)**2)
      SQCER=0.
      DO 1 I=1,NT
         SQCER=SQCER+CABS(SIRCH(I)-SHUK(1,I))**2
1     CONTINUE
      IF (ITIME.LE.ITRIT) GOTO 3
      PERRI=FLOAT(ITIME-ITRIT)
      AVSQER=AVSQER+(SQERR-AVSQER)/PERRI
      AVSIRE=AVSIRE+(CABS(SIRER)**2-AVSIRE)/PERRI
      AVCER=AVCER+(SQCER-AVCER)/PERRI
3     DO 2 I=1,NT
         IF (.NOT.TRAIN) THEN
         XY=CONSIG(KINP(IDID+2-I))
         ELSE
         XY=CONSIG(INP(IDID+NT+1-I))
         ENDIF
         SIRCH(I)=SIRCH(I)+DELTA*SIRER*CONJG(XY)
2     CONTINUE
      RETURN
      END
```

<u>SUBROUTINE DTAPS</u>

```
      SUBROUTINE DTAPS
      COMPLEX SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,
     +A(20),W(20),V(20),HLAM,A1,B1,D0
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      IF (SIRCH(1).EQ.(0.,0.)) RETURN
      HLAM=1./CONJG(SIRCH(1))
      A(1)=2.*SIGM*HLAM
      DO 1 I=1,(LANT-1)
         A(I+1)=(0.,0.)
         DO 2 K=MAX0(0,I-NT+1),I-1
            A(I+1)=A(I+1)-CONJG(SIRCH(I-K+1))*A(K+1)
2        CONTINUE
         A(I+1)=A(I+1)*HLAM
1     CONTINUE
```

```
      A(1)=A(1)+SIRCH(1)
      HLAM=1./A(1)
      W(1)=A(2)*HLAM
      V(1)=SIRCH(2)*HLAM
      D0=A(1)
      A1=W(1)
      B1=V(1)
      DO 3 K=1,LANT-2
         D0=(1.-A1*B1)*D0
         A1=A(K+2)
         DO 4 I=1,K
            A1=A1-A(I+1)*W(K-I+1)
4        CONTINUE
         HLAM=1./D0
         A1=A1*HLAM
         B1=(0.,0.)
         IF (K.LE.NT-2) B1=SIRCH(K+2)
         DO 5 I=1,MIN0(K,NT-1)
            B1=B1-SIRCH(I+1)*V(K-I+1)
5        CONTINUE
         B1=B1*HLAM
         DO 6 I=1,K
            HLAM=V(K-I+1)-B1*W(I)
            W(I)=W(I)-A1*V(K-I+1)
            V(K-I+1)=HLAM
6        CONTINUE
         W(K+1)=A1
         V(K+1)=B1
3     CONTINUE
      D0=(1.-A1*B1)*D0
      ALF(1)=1./D0
      DO 7 I=1,LANT-1
         ALF(I+1)=-ALF(1)*W(I)
7     CONTINUE
      DO 8 I=2,NT
         EST(I)=(0.,0.)
         DO 9 K=1,MIN0(LANT,NT-I+1)
            EST(I)=EST(I)+ALF(K)*SIRCH(K+I-1)
```

```
9        CONTINUE
8     CONTINUE
      RETURN
      END


      SUBROUTINE IDECOD(IRE1,IRE0,IX,IY)
      COMMON/CONSIY/NINQ,NSPA
      IQUA1=IRE1/NINQ
      IQUA0=IRE0/NINQ
      IQUAD=MOD((4+IQUA1-IQUA0),4)
      LOK=MOD(IRE1,NINQ)
      IY=2*(LOK/NSPA)+1
      IX=2*MOD(LOK,NSPA)+1
      IF (IQUAD.EQ.1) THEN
      I=IX
      IX=-IY
      IY=I
      ELSE IF (IQUAD.EQ.2) THEN
      IX=-IX
      IY=-IY
      ELSE IF (IQUAD.EQ.3) THEN
      I=IX
      IX=IY
      IY=-I
      ENDIF
      RETURN
      END


      COMPLEX FUNCTION CGAUSS(ISEED)
      CALL RANSET(ISEED)
      THETE=6.283185308*RANF()
1     UNIF=RANF()
      IF (UNIF.LE.0.0) GOTO 1
      CALL RANGET(ISEED)
      RADIUS=SQRT(-2.0*LOG(UNIF))
      CGAUSS=RADIUS*CMPLX(COS(THETE),SIN(THETE))
      RETURN
      END


      COMPLEX FUNCTION CONSIG(N)
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/CONSIY/NINQ,NSPA
      IF (N.NE.-1) GOTO 1
      CONSIG=(0.,0.)
      RETURN
1     LOK=MOD(N,NINQ)
      LOKI=2*(LOK/NSPA)+1
      LOKR=2*MOD(LOK,NSPA)+1
      IQUAD=N/NINQ
      X=FLOAT(LOKR)
      Y=FLOAT(LOKI)
      IF (IQUAD.EQ.1) THEN
      R=X
      X=-Y
      Y=R
      ELSE IF (IQUAD.EQ.2) THEN
      X=-X
      Y=-Y
      ELSE IF (IQUAD.EQ.3) THEN
      R=X
      X=Y
      Y=-R
      ENDIF
      CONSIG=CMPLX(X,Y)/PIFF
      RETURN
      END


      FUNCTION MDECON(IX,IY)
      COMMON/CONSIY/NINQ,NSPA
      IQUAD=1
      IF (ISIGN(1,IX).EQ.ISIGN(1,-IY)) IQUAD=2
      IQUAD=IQUAD-ISIGN(1,IY)
      IDX=IX
      IDY=IY
      IF (IQUAD.EQ.1) THEN
      I=IDX
      IDX=IDY
      IDY=-I
      ELSE IF (IQUAD.EQ.2) THEN
      IDX=-IDX
      IDY=-IDY
      ELSE IF (IQUAD.EQ.3) THEN
      I=IDX
      IDX=-IDY
      IDY=I
      ENDIF
      MDECON=IQUAD*NINQ+NSPA*(IDY-1)/2+(IDX-1)/2
      RETURN
      END


      SUBROUTINE SRKAL1(AR1,REVISE,TRAIN)
      COMPLEX AR1,ERRID,SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,UUU,
     +RDASH,CONSIG,FFF(30),GGG(30),HLAMDA,BETA1,XY
      DIMENSION AAA(30)
      LOGICAL REVISE,TRAIN
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
      COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
      COMMON/KALSRK/UUU(410),DDD(30),EPSIL,QUE,HQUEP
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      RDASH=(0.,0.)
      IF (.NOT.TRAIN) THEN
      DO 11 I=1,NT
          XY=CONSIG(KINP(IDID+2-I))
          RDASH=RDASH+XY*EST(I)
 11    CONTINUE
      ELSE
      DO 12 I=1,NT
         RDASH=RDASH+CONSIG(INP(IDID+NT+1-I))*EST(I)
12    CONTINUE
      ENDIF
      ERRID=AR1-RDASH
      SQERR=SVAR*(CABS(ERRID)**2)
      IF (ITIME.LE.ITRIT) GOTO 10
      AVSQER=AVSQER+(SQERR-AVSQER)/FLOAT(ITIME-ITRIT)
10    FFF(1)=CONJG(RALF(IDID+1))
      GGG(1)=DDD(1)*FFF(1)
      XY=GGG(1)*CONJG(FFF(1))
      AAA(1)=EPSIL+REAL(XY)
      MM=0
      DO 1 J=2,LANT
         FFF(J)=CONJG(RALF(IDID+J))
         DO 2 I=1,(J-1)
            MM=MM+1
            FFF(J)=FFF(J)+UUU(MM)*CONJG(RALF(IDID+I))
2        CONTINUE
         GGG(J)=DDD(J)*FFF(J)
         XY=GGG(J)*CONJG(FFF(J))
         AAA(J)=AAA(J-1)+REAL(XY)
1     CONTINUE
      DO 3 J=(LANT+1),(LANT+NT-1)
         LL=J-LANT-1
         IF (.NOT.TRAIN) THEN
         XY=CONSIG(KINP(IDID-LL))
         ELSE
         XY=CONSIG(INP(IDID+NT-1-LL))
         ENDIF
         FFF(J)=CONJG(XY)
         DO 4 I=1,LANT
            MM=MM+1
            FFF(J)=FFF(J)+UUU(MM)*CONJG(RALF(IDID+I))
4        CONTINUE
         LL=-1
         DO 5 I=(LANT+1),(J-1)
            MM=MM+1
```

```
              LL=LL+1
              IF (.NOT.TRAIN) THEN
              XY=CONSIG(KINP(IDID-LL))
              ELSE
              XY=CONSIG(INP(IDID+NT-1-LL))
              ENDIF
              FFF(J)=FFF(J)+UUU(MM)*CONJG(XY)
5         CONTINUE
          GGG(J)=DDD(J)*FFF(J)
          XY=GGG(J)*CONJG(FFF(J))
          AAA(J)=AAA(J-1)+REAL(XY)
3     CONTINUE
      IF (REVISE) THEN
      HHTT=QUE*AAA(LANT+NT-1)
      ELSE
      HHTT=0.
      ENDIF
      CAPPA=1./(AAA(1)+HHTT)
      DDD(1)=DDD(1)*HQUEP*(EPSIL+HHTT)*CAPPA
      MM=0
      DO 6 J=2,(LANT+NT-1)
          BETA=AAA(J-1)+HHTT
          HLAMDA=FFF(J)*CAPPA
          CAPPA=1./(AAA(J)+HHTT)
          DDD(J)=DDD(J)*HQUEP*BETA*CAPPA
          DO 7 I=1,(J-1)
              MM=MM+1
              BETA1=UUU(MM)
              UUU(MM)=BETA1-CONJG(GGG(I))*HLAMDA
              GGG(I)=GGG(I)+GGG(J)*CONJG(BETA1)
7         CONTINUE
6     CONTINUE
      BETA1=-ERRID/AAA(LANT+NT-1)
      DO 8 J=1,LANT
          ALF(J)=ALF(J)+GGG(J)*BETA1
8     CONTINUE
      DO 9 J=2,NT
          EST(J)=EST(J)-GGG(LANT+J-1)*BETA1
```

```
9    CONTINUE
     RETURN
     END


     SUBROUTINE CONVER(VERG,ENCE)
     COMPLEX ERRID,SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,UUU
     DIMENSION VERG(61),ENCE(61)
     COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
     COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
     COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
     COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
     COMMON/KALSRK/UUU(410),DDD(30),EPSIL,QUE,HQUEP
     COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
     IF (ITIME.EQ.ITRIT) THEN
     DO 1 I=1,LANT
         ALF(I)=(0.,0.)
1    CONTINUE
     SIRCH(1)=(0.,0.)
     DO 2 I=2,NT
         EST(I)=(0.,0.)
         SIRCH(I)=(0.,0.)
2    CONTINUE
     J=(LANT+NT-1)*(LANT+NT-2)
     DO 3 I=1,J/2
         UUU(I)=(0.,0.)
3    CONTINUE
     DO 4 I=1,(LANT+NT-1)
         DDD(I)=1.
4    CONTINUE
     RETURN
     ENDIF
     I=ITIME-ITRIT
     B1=LOG10(SQCER+1.1E-100)
     B2=LOG10(SQISI+SQNOI+1.1E-100)
     VERG(I)=VERG(I)+(1./30.)*B1
     ENCE(I)=ENCE(I)+(1./30.)*B2
     RETURN
```

```
      END


      SUBROUTINE DTAPS1
      COMPLEX SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,A(20,20),V(20),B2
      DIMENSION D(20)
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      DO 16 I=1,LANT
         DO 17 J=1,I
            A(J,I)=(0.,0.)
17       CONTINUE
16    CONTINUE
      DO 3 M=1,LANT
         A(M,M)=A(M,M)+2.*SIGM
         DO 4 I=M,MIN0((M+NT-1),LANT)
            DO 5 J=M,I
            A(J,I)=A(J,I)+CONJG(SHUK(I,(I-M+1)))*SHUK(J,(J-M+1))
5           CONTINUE
4        CONTINUE
3     CONTINUE
      D(1)=REAL(A(1,1))
      DO 6 I=2,LANT
         B1=REAL(A(I,I))
         DO 7 J=1,(I-1)
            B2=A(J,I)
            DO 8 K=1,(J-1)
               B2=B2-V(K)*CONJG(A(J,K))
8           CONTINUE
            V(J)=B2
            A(I,J)=B2/D(J)
            B1=B1-(CABS(B2)**2)/D(J)
7        CONTINUE
         D(I)=B1
         IF (D(I).EQ.0.) RETURN
6     CONTINUE
```

```
      DO 14 I=1,LANT
          IF (I.LE.NT) THEN
          B2=CONJG(SHUK(I,I))
          ELSE
          B2=(0.,0.)
          ENDIF
          DO 15 K=1,(I-1)
              B2=B2-A(I,K)*V(K)
15        CONTINUE
          V(I)=B2
14    CONTINUE
      DO 19 I=LANT,1,-1
          B2=V(I)/D(I)
          DO 20 K=(I+1),LANT
              B2=B2-CONJG(A(K,I))*ALF(K)
20        CONTINUE
          ALF(I)=B2
19    CONTINUE
      SIRCH(1)=SHUK(1,1)
      DO 9 I=2,NT
          EST(I)=(0.,0.)
          DO 13 J=1,MIN0(LANT,(NT-I+1))
              EST(I)=EST(I)+ALF(J)*SHUK(J,(J+I-1))
13        CONTINUE
          SIRCH(I)=SHUK(1,I)
9     CONTINUE
      RETURN
      END
```

<u>SUBROUTINE EVAL1</u>

```
      COMPLEX ERRID,SIRCH,SIRER,SHUK,EST,RALF,ALF,RVEC,XY
      DIMENSION CLASS1(50),CLASS2(100),CLASS3(150)
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/DATSYM/IECNT(5),KINP(22),KINP1(21),IDECS(10),INP(50)
      COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      COMMON/DFEFIL/RALF(40),ALF(20),RVEC(21)
      COMMON/MISC1/ITIME,ITRIT,LANT,ISED1,ISED2,KADEL,DELT0,DAMSE,DELTA
      SQERR=2.*SIGM*CABS(ALF(1))/CABS(SHUK(1,1))
      IF (ITIME.LE.ITRIT) RETURN
      PERRI=FLOAT(ITIME-ITRIT)
      AVSQER=AVSQER+(SQERR-AVSQER)/PERRI
      IIY=2*LANT-1
      IIX=IIY
      XX=0.
      B1=0.
      MM=0
      DO 2 I=1,(LANT-1)
         XY=(0.,0.)
         DO 3 J=(I+1),MIN0(LANT,(NT+I))
            XY=XY+ALF(J)*SHUK(J,(J-I))
3        CONTINUE
         C1=ABS(REAL(XY))
         C2=ABS(AIMAG(XY))
         CLASS1(MM+1)=C1
         CLASS1(MM+2)=C2
         DO 12 J=1,2
            B2=4./(2.**J)
            CLASS2(2*MM+2*J-1)=C1*B2
            CLASS2(2*MM+2*J)=C2*B2
12       CONTINUE
         DO 13 J=1,3
            B2=8./(2.**J)
            CLASS3(3*MM+2*J-1)=C1*B2
            CLASS3(3*MM+2*J)=C2*B2
13       CONTINUE
         MM=MM+2
         XX=XX+CABS(XY)**2
         B1=B1+CABS(ALF(I))**2
2     CONTINUE
      SQISI=XX
      B1=B1+CABS(ALF(LANT))**2
      SQNOI=2.*SIGM*B1
      XY=(0.,0.)
      DO 4 I=1,MIN0(LANT,NT)
```

```
            XY=XY+ALF(I)*SHUK(I,I)
4       CONTINUE
        XY=XY-1.
        A1=REAL(XY)
        C2=ABS(AIMAG(XY))
        CLASS1(MM+1)=C2
        DO 14 J=1,2
            B2=4./(2.**J)
            CLASS2(2*MM+J)=C2*B2
14      CONTINUE
        DO 15 J=1,3
            B2=8./(2.**J)
            CLASS3(3*MM+J)=C2*B2
15      CONTINUE
        SQISI=SQISI+CABS(XY)**2
        AVISI=AVISI+(SQISI-AVISI)/PERRI
        AVNOI=AVNOI+(SQNOI-AVNOI)/PERRI
        DO 5 I=1,(IIY-1)
            B1=0.
            DO 6 J=I,IIY
                IF (B1.GT.CLASS1(J)) GOTO 6
                B1=CLASS1(J)
                MM=J
6           CONTINUE
            IF (B1.EQ.0.) THEN
            IIX=I-1
            GOTO 34
            ENDIF
            CLASS1(MM)=CLASS1(I)
            CLASS1(I)=B1
5       CONTINUE
        IF (CLASS1(IIY).EQ.0.) IIX=IIY-1
34      DO 16 I=1,(2*IIY-1)
            B1=0.
            DO 17 J=I,2*IIY
                IF (B1.GT.CLASS2(J)) GOTO 17
                B1=CLASS2(J)
                MM=J
```

```
17          CONTINUE
          CLASS2(MM)=CLASS2(I)
          CLASS2(I)=B1
16    CONTINUE
      DO 18 I=1,(3*IIY-1)
          B1=0.
          DO 19 J=I,3*IIY
              IF (B1.GT.CLASS3(J)) GOTO 19
              B1=CLASS3(J)
              MM=J
19          CONTINUE
          CLASS3(MM)=CLASS3(I)
          CLASS3(I)=B1
18    CONTINUE
      IIY=IIX
      C1=1./SQRT(2.*SQNOI)
      F1=0.
      F2=0.
      D1=1.+A1
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      IF (IIY.EQ.0) THEN
      F2=F2+E1
      GOTO 35
      ENDIF
      C2=0.5
      F2=F2+C2*E1
      DO 21 J=1,(IIY-1)
          D1=D1-CLASS1(J)
          E1=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          C2=C2/2.
          F2=F2+C2*E1
21    CONTINUE
      D1=D1-CLASS1(IIY)
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      F2=F2+C2*E1
      D1=2.*(1.+A1)-D1
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      F1=F1+E1+F2
35     PEULB(1)=F1/2.
      PEULB(2)=F2
      PEULB(3)=PEULB(3)+(PEULB(1)-PEULB(3))/PERRI
      PEULB(4)=PEULB(4)+(PEULB(2)-PEULB(4))/PERRI
      C1=1./SQRT(10.*SQNOI)
      F1=0.
      F2=0.
      DO 22 I=2,2
          K=2*I-5
          D1=1.-FLOAT(K)*A1
          D2=1.+FLOAT(K)*A1
          E1=0.
          E2=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          B2=D2*C1
          IF (B2.LE.25.) E2=ERFC(B2)
          IF (IIY.EQ.0) THEN
          F2=F2+E1+E2
          GOTO 22
          ENDIF
          C2=0.5
          F2=F2+C2*(E1+E2)
          DO 23 J=1,(2*IIY-1)
              D1=D1-CLASS2(J)
              D2=D2-CLASS2(J)
              E1=0.
              E2=0.
              B2=D1*C1
              IF (B2.LE.25.) E1=ERFC(B2)
              B2=D2*C1
              IF (B2.LE.25.) E2=ERFC(B2)
              C2=C2/2.
              F2=F2+C2*(E1+E2)
23          CONTINUE
          D1=D1-CLASS2(2*IIY)
          D2=D2-CLASS2(2*IIY)
          E1=0.
          E2=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          B2=D2*C1
          IF (B2.LE.25.) E2=ERFC(B2)
          F2=F2+C2*(E1+E2)
          D1=2.*(1.-FLOAT(K)*A1)-D1
          D2=2.*(1.+FLOAT(K)*A1)-D2
          E1=0.
          E2=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          B2=D2*C1
          IF (B2.LE.25.) E2=ERFC(B2)
          F1=F1+E1+E2
22    CONTINUE
      D1=1.+3.*A1
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      IF (IIY.EQ.0) THEN
      F2=F2+E1
      GOTO 36
      ENDIF
      C2=0.5
      F2=F2+C2*E1
      DO 24 J=1,(2*IIY-1)
          D1=D1-CLASS2(J)
          E1=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          C2=C2/2.
          F2=F2+C2*E1
24     CONTINUE
       D1=D1-CLASS2(2*IIY)
       E1=0.
       B2=D1*C1
       IF (B2.LE.25.) E1=ERFC(B2)
       F2=F2+C2*E1
       D1=2.*(1.+3.*A1)-D1
       E1=0.
       B2=D1*C1
       IF (B2.LE.25.) E1=ERFC(B2)
       F1=F1+E1+F2
36     PEULB(5)=F1/4.
       PEULB(6)=F2/2.
       PEULB(7)=PEULB(7)+(PEULB(5)-PEULB(7))/PERRI
       PEULB(8)=PEULB(8)+(PEULB(6)-PEULB(8))/PERRI
       C1=1./SQRT(42.*SQNOI)
       F1=0.
       F2=0.
       DO 28 I=2,4
          K=2*I-9
          D1=1.-FLOAT(K)*A1
          D2=1.+FLOAT(K)*A1
          E1=0.
          E2=0.
          B2=D1*C1
          IF (B2.LE.25.) E1=ERFC(B2)
          B2=D2*C1
          IF (B2.LE.25.) E2=ERFC(B2)
          IF (IIY.EQ.0) THEN
          F2=F2+E1+E2
          GOTO 28
          ENDIF
          C2=0.5
          F2=F2+C2*(E1+E2)
          DO 29 J=1,(3*IIY-1)
```

```
            D1=D1-CLASS3(J)
            D2=D2-CLASS3(J)
            E1=0.
            E2=0.
            B2=D1*C1
            IF (B2.LE.25.) E1=ERFC(B2)
            B2=D2*C1
            IF (B2.LE.25.) E2=ERFC(B2)
            C2=C2/2.
            F2=F2+C2*(E1+E2)
29          CONTINUE
         D1=D1-CLASS3(3*IIY)
         D2=D2-CLASS3(3*IIY)
         E1=0.
         E2=0.
         B2=D1*C1
         IF (B2.LE.25.) E1=ERFC(B2)
         B2=D2*C1
         IF (B2.LE.25.) E2=ERFC(B2)
         F2=F2+C2*(E1+E2)
         D1=2.*(1.-FLOAT(K)*A1)-D1
         D2=2.*(1.+FLOAT(K)*A1)-D2
         E1=0.
         E2=0.
         B2=D1*C1
         IF (B2.LE.25.) E1=ERFC(B2)
         B2=D2*C1
         IF (B2.LE.25.) E2=ERFC(B2)
         F1=F1+E1+E2
28      CONTINUE
      D1=1.+7.*A1
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      IF (IIY.EQ.0) THEN
      F2=F2+E1
      GOTO 37
      ENDIF
      C2=0.5
      F2=F2+C2*E1
      DO 30 J=1,(3*IIY-1)
         D1=D1-CLASS3(J)
         E1=0.
         B2=D1*C1
         IF (B2.LE.25.) E1=ERFC(B2)
         C2=C2/2.
         F2=F2+C2*E1
30    CONTINUE
      D1=D1-CLASS3(3*IIY)
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      F2=F2+C2*E1
      D1=2.*(1.+7.*A1)-D1
      E1=0.
      B2=D1*C1
      IF (B2.LE.25.) E1=ERFC(B2)
      F1=F1+E1+F2
37    PEULB(9)=F1/8.
      PEULB(10)=F2/4.
      PEULB(11)=PEULB(11)+(PEULB(9)-PEULB(11))/PERRI
      PEULB(12)=PEULB(12)+(PEULB(10)-PEULB(12))/PERRI
      RETURN
      END


      SUBROUTINE EVAL2
      COMPLEX ERRID,SIRCH,SIRER,SHUK,EST
      DIMENSION CLASS1(50),CLASS2(100),CLASS3(150)
      COMMON/MISC/NT,NSL,IDID,IL,IL1,SNR0,SIGM,PIFF,SVAR
      COMMON/PICO/ERRID,SQERR,AVSQER,SQISI,AVISI,SQNOI,AVNOI,PEULB(12)
      COMMON/GLUM/SIRCH(10),SIRER,AVSIRE,SQCER,AVCER,SHUK(20,10),EST(10)
      CHEN=0.
      DO 1 I=1,NT
         CHEN=CHEN+CABS(SHUK(1,I))**2
1     CONTINUE
```

```
SQERR=2.*SIGM/CHEN
SQISI=2.*SIGM/(CHEN+2.*SIGM)
IF (ITIME.LE.ITRIT) RETURN
PERRI=FLOAT(ITIME−ITRIT)
AVSQER=AVSQER+(SQERR−AVSQER)/PERRI
AVISI=AVISI+(SQISI−AVISI)/PERRI
P1=1./(2.*SQERR)
P1=SQRT(P1)
E1=0.
IF (P1.LE.25.) E1=ERFC(P1)
PERRI=FLOAT(ITIME−ITRIT)
PEULB(1)=E1
PEULB(2)=PEULB(2)+(PEULB(1)−PEULB(2))/PERRI
P1=1./(10.*SQERR)
P1=SQRT(P1)
E1=0.
IF (P1.LE.25.) E1=ERFC(P1)
PEULB(3)=1.5*E1
PEULB(4)=PEULB(4)+(PEULB(3)−PEULB(4))/PERRI
P1=1./(42.*SQERR)
P1=SQRT(P1)
E1=0.
IF (P1.LE.25.) E1=ERFC(P1)
PEULB(5)=1.75*E1
PEULB(6)=PEULB(6)+(PEULB(5)−PEULB(6))/PERRI
RETURN
END
```

REFERENCES

[1] Monsen, P.. "Fading Channel Communications", IEEE Communications Magazine, p. 16–25, January 1980.

[2] Maslin, N. M.. "High data rate transmissions over h.f. links", The Radio and Electronic Engineer, vol. 52, No. 2, p. 75–87, February 1982.

[3] Proakis, J. G.. "Digital Communications", International Student Edition, McGraw-Hill, 1983.

[4] "HF Ionospheric Channel Simulators", CCIR (Comitè Consultatif International des Radio Communications, or International Radio Consultative Committee) Report 549-1 (1974–78), 1982.

[5] Davies, K.. "Ionospheric Radio Propagation", Dover, New York 1966.

[6] Mosier, R. R.; Calbaugh, R. G.. "Kineplex, a bandwidth-efficient binary transmission system", AIEE Transactions on Communications and Electronics, Part 1, vol. 76, p. 723–728, 1958.

[7] Porter, G. C.. "Error Distribution and Diversity Performance of a Frequency-Differential PSK HF Modem", IEEE Transactions on Communication Technology, vol. COM-16, No. 4, p. 567–575, August 1968.

[8] Zimmerman, M. S.; Kirsch, A. L.. "The AN/GSC-10 (KATHRYN) Variable Rate Data Modem for HF Radio", IEEE Transactions on Communication Technology, vol. COM-15, No. 2, p. 197–204, April 1967.

[9] Price, R.; Green, P. E.. "A Communication Technique for Multipath Channels", Proceedings of the IRE, vol. 46, p. 555–570, March 1958.

[10] Di Toro, M. J.. "Communication in time-frequency spread media using adaptive equalization", Proceedings of the IEEE, vol. 56, p.1653–1679, 1968.

[11] Clark, A. P.; McVerry, F.. "Performance of 2400 bits/s serial and parallel modems over an HF channel simulator", IERE Conference on Digital Processing of Signals in Communication, Proceedings No. 49, Loughborough, England, p. 167–179, April 1981.

[12] Hodgkiss, W.; Turner, L. F.; Pennington, J.. "Serial data transmission over HF radio links", IEE Proceedings, vol. 131, Pt. F, No. 2, p. 107–116, April 1984.

[13] Darnell, M.. "Medium-speed digital data transmission over HF channels", IEEE/IEE/IERE Conference on Signal Processing, Loughborough, England, 1977.

[14] de Pedro, H.; Hsu, F.; Giordano, A.; Proakis, J.. "Signal Design for High-Speed Serial Transmission on Fading Dispersive Channels", NTC '78, Conference Record of the IEEE 1978 National Telecommunications Conference, Pt. II, Birmingham, AL, USA, 3–6 December 1978 (New York, USA: IEEE 1978), p. 27.4/1–4.

[15] Hsu, F. M.; Giordano, A. A.; de Pedro, H. E.; Proakis, J. G.. "Adaptive Equalization Techniques for High-Speed Transmission on Fading Dispersive HF Channels", NTC '80, IEEE 1980 National Telecommunications Conference, Houston, TX, USA, 30 Nov.–4 Dec. 1980 (New York, USA: IEEE 1980), p. 58.1/1–7.

[16] Hsu, F. M.. "Square Root Kalman Filtering for High-Speed Data Received over Fading Dispersive HF Channels", IEEE Transactions on Information Theory, vol. IT-28, No. 5, p. 753–763, September 1982.

[17] Clark, A. P.; Asghar, S. M.. "Detection of digital signals transmitted over a known time-varying channel", IEE Proceedings, vol. 128, Pt. F, No. 3, p. 167–174, June 1981.

[18] Clark, A. P.; Najdi, H. Y.. "Detection process of a 9600 bit/s serial modem for HF radio links", IEE Proceedings, vol. 130, Pt. F, No. 5, p. 368–376, August 1983.

[19] Lucky; Salz; Weldon. "Principles of Data Communication", McGraw-Hill, 1968.

[20] Sharpe, J. T. L.. "Techniques for High-Speed Data Transmission over Voice Channels", Electrical Communication, vol. 46, No. 1, p. 24–31, 1971.

[21] Clark, A. P.. "Advanced Data-Transmission Systems", Pentech Press, 1977.

[22] Ericson, T.. "Structure of optimum receiving filters in data transmission systems", IEEE Transactions on Information Theory (Correspondence), vol. IT-17, p. 352–353, May 1971.

[23] Forney, G. D.. "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference", IEEE Transactions on Information Theory, vol. IT-18, No. 3, p. 363–378, May 1972.

[24] Qureshi, S. U. H.. "Adaptive Equalization", Proceedings of the IEEE, vol. 73, No. 9, p. 1349–1387, September 1985.

[25] Mueller, M. S.; Salz, J.. "A Unified Theory of Data-Aided Equalization", Bell System Technical Journal, vol. 60, No. 9, p. 2023–2038, November 1981.

[26] Salz, J.. "Optimum Mean-Square Decision Feedback Equalization", Bell System Technical Journal, vol. 52, No. 8, p. 1341–1373, October 1973.

[27] Messerschmitt, D. G.. "A Geometric Theory of Intersymbol Interference", Bell System Technical Journal, vol. 52, No. 9, Part I: p. 1483–1519; Part II: p. 1521–1539, November 1973.

[28] Monsen, P.. "Feedback Equalization for Fading Dispersive Channels", IEEE Transactions on Information Theory, vol. IT-17, p. 56–64, January 1971.

[29] Belifore, C. A.; Park, J. H.. "Decision Feedback Equalization", Proceedings of the IEEE, vol. 67, No. 8, p. 1143–1156, August 1979.

[30] Ungerboeck, G.. "Adaptive Maximum-Likelihood Receiver for Carrier-Modulated Data-Transmission Systems", IEEE Transactions on Communications, vol. COM-22, No. 5, p. 624–636, May 1974.

[31] Forney, G. D.. "The Viterbi Algorithm", Proceedings of the IEEE, vol. 61, No. 3, p. 268–278, March 1973.

[32] Anderson, Ib N.. "Sample-Whitened Matched Filters", IEEE Transactions on Information Theory, vol. IT-19, No. 5, p. 653–660, September 1973.

[33] Foschini, G. J.. "Performance bound for maximum likelihood reception of digital data", IEEE Transactions on Information Theory, vol. IT-21, p. 47–50, January 1975.

[34] Qureshi, S. U. H; Newhall, E. E.. "An Adaptive Receiver for Data Transmission over Time-Dispersive Channels", IEEE Transactions on Information Theory, vol. IT-19, No. 4, p. 448–457, July 1973.

[35] Beare, C. T.. "The Choice of the Desired Impulse Response in Combined Linear-Viterbi Algorithm Equalizers", IEEE Transactions on Communications, vol. COM-26, No. 8, p. 1301–1307, August 1978.

[36] Falconer, D. D.; Magee, F. R.. "Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation", Bell System Technical Journal, vol. 52, No. 9, p. 1541–1562, November 1973.

[37] Fagan, A. D.; O'Keane, F. D.. "Performance comparison of detection methods derived from maximum-likelihood sequence estimation", IEE Proceedings, vol. 133, Pt. F, No. 6, p. 535–542, October 1986.

[38] Foschini, G. J.. "A Reduced State Variant of Maximum Likelihood Sequence Detection Attaining Optimum Performance for High Signal-to-Noise Ratios", IEEE Transactions on Information Theory, vol. IT-23, No. 5, p. 605–609, September 1977.

[39] Clark, A. P.; Harvey, J. D.; Driscoll, J. P.. "Near-maximum-likelihood detection processes for distorted digital signals", Radio and Electronic Engineer, vol. 48, No. 6, p. 301–309, June 1978.

[40] Clark, A. P.; Kwong, C. P.; Harvey, J. D.. "Detection processes for severely distorted digital signals", Electronic Circuits and Systems, vol. 3, No. 1, p. 27–37, January 1979.

[41] Clark, A. P.; Fairfield, M. J.. "Detection processes for a 9600 bit/s modem", Radio and Electronic Engineer, vol. 51, No. 9, p. 455–465, September 1981.

[42] Clark, A. P.; Najdi, H. Y.; Fairfield, M. J.. "Data transmission at 19.2 kbit/s over telephone circuits", Radio and Electronic Engineer, vol. 53, No. 4, p. 157–166, April 1983.

[43] Clark, A. P.; Clayden, M.. "Pseudobinary Viterbi detector", IEE Proceedings, vol. 131, Pt. F, No. 2, p. 208–218, April 1984.

[44] Price, R.. "Nonlinearly feedback-equalized PAM vs. capacity for noisy filter channels", Proceedings of the 1972 IEEE International Conference on Communications, ICC '72 (Philadelphia, PA), p. 22-12 to 22-17, June 1972.

[45] Magee, F. R.. "A Comparison of Compromise Viterbi Algorithm and Standard Equalization Techniques Over Band-Limited Channels", IEEE Transactions on Communications, vol. COM-23, No. 3, p. 361–367, March 1975.

[46] Falconer, D. D.. "Application of Passband Decision Feedback Equalization in Two-Dimensional Data Communication Systems", IEEE Transactions on Communications, vol. COM-24, No. 10, p. 1159–1165, October 1976.

[47] Duttweiler, D. L.; Mazo, J. E.; Messerschmitt, D. G.. "An Upper Bound on the Error Probability in Decision Feedback Equalization", IEEE Transactions on Information Theory, vol. IT-20, No. 4, p. 490–497, July 1974.

[48] Proakis, J. G.. "Adaptive non-linear filtering techniques for data transmission", IEEE Symposium on Adaptive Processes, Decision and Control, p. XV.2.1–5, 1970.

[49] Falconer, D. D.; Magee, F. R.. "Evaluation of decision feedback equalization and Viterbi algorithm detection for voiceband data transmission", IEEE Transactions on Communications, vol. COM-24, Part I: p. 1130–1139 (October); Part II: p. 1238–1245 (November), 1976.

[50] Lyon, D. L.. "Envelope-Derived Timing Recovery in QAM and SQAM Systems", IEEE Transactions on Communications, vol. COM-23, p. 1327–1331, November 1975.

[51] Mengali, U.. "Synchronization of QAM signals in the presence of ISI", IEEE Transactions on Aerospace and Electronic Systems, vol. AES-12, p. 556–560, September 1976.

[52] Franks, L. E.. "Carrier and Bit Synchronization in Data Communication—A Tutorial Review", IEEE Transactions on Communications, vol. COM-28, No. 8, p. 1107–1121, August 1980.

[53] Meyers, M. H.; Franks, L. E.. "Joint Carrier Phase and Symbol Timing Recovery for PAM Systems", IEEE Transactions on Communications, vol. COM-28, No. 8, p. 1121–1129, August 1980.

[54] Moeneclaey, M.. "Synchronization problems in PAM Systems", IEEE Transactions on Communications, vol. COM-28, No. 8, p. 1130–1136, August 1980.

[55] Hodgkiss, W.; Turner, L. F.. "Practical equalization and synchronization strategies for use in serial data transmission over h.f. channels", The Radio and Electronic Engineer, vol. 53, No. 4, p. 141–146, April 1983.

[56] Nicholson, G.; Norton, J. P.. "Kalman Filter Equalization for a Time-Varying Communication Channel", Australian Telecommunication Research, vol. 13, No. 1, p. 3–12, 1979.

[57] Ungerboeck, G.. "Theory on the Speed of Convergence in Adaptive Equalizers for Digital Communication", IBM Journal of Research and Development, vol. 16, p. 546–555, November 1972.

[58] Widrow, B.; McCool, J. M.; Larimore, M. G.; Johnson, C. R.. "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter", Proceedings of the IEEE, vol. 64, No. 8, p. 1151–1162, August 1976.

[59] Widrow, B.; Walach, E.. "On the Statistical Efficiency of the LMS Algorithm with Nonstationary Inputs", IEEE Transactions on Information Theory, vol. IT-30, No. 2, p. 211–221, March 1984.

[60] Proakis, J. G.; Miller, J. H.. "An Adaptive Receiver for Digital Signaling Through Channels With Intersymbol Interference", IEEE Transactions on Information Theory, vol. IT-15, No. 4, p. 484–497, July 1969.

[61] Godard, D.. "Channel Equalization Using a Kalman Filter for Fast Data Transmission", IBM Journal of Research and Development, vol. 18, p. 267–273, May 1974.

[62] Mueller, M. S.. "Least-Squares Algorithms for Adaptive Equalizers", Bell System Technical Journal, vol. 60, No. 8, p. 1905–1925, October 1981.

[63] Mueller, M. S.. "On the Rapid Initial Convergence of Least-Squares Equalizer Adjustment Algorithms", Bell System Technical Journal, vol. 60, No. 10, p. 2345–2358, December 1981.

[64] Giordano, A. A.; Hsu, F. M.. "Least Square Estimation with Applications to Digital Signal Processing", Wiley-Interscience Publication, John Wiley & Sons, 1985.

[65] Falconer, D. D.; Ljung, L.. "Application of Fast Kalman Estimation to Adaptive Equalization", IEEE Transactions on Communications, vol. COM-26, p. 1439–1446, October 1978.

[66] Clark, A. P.; McVerry, F.. "Channel estimation for an HF radio link", IEE Proceedings, vol. 128, Pt. F, No. 1, p. 33–42, February 1981.

[67] Watterson, C. C.; Juroshek, J. R.; Bensema, W. D.. "Experimental Confirmation of an HF Channel Model", IEEE Transactions on Communication Technology, vol. COM-18, No. 6, p. 792–803, December 1970.

[68] Perl, J. M.; Kagan, D.. "Real-Time HF Channel Parameter Estimation", IEEE Transactions on Communications, vol. COM-34, No. 1, p. 54–58, January 1986.

[69] Chang, R. W.; Srinivasagopalan, R.. "Carrier Recovery for Data Communication Systems with Adaptive Equalization", IEEE Transactions on Communications, vol. COM-28, No. 8, p. 1142–1153, August 1980.

[70] Falconer, D. D.. "Jointly adaptive equalization and carrier recovery in two-dimensional digital communication systems", Bell System Technical Journal, vol. 55, p. 316–334, March 1976.

[71] Ling, F.; Proakis, J. G.. "Adaptive Lattice Decision-Feedback Equalizers—Their Performance and Application to Time-Variant Multipath Channels", IEEE Transactions on Communications, vol. COM-33, No. 4, p. 348–356, April 1985.

[72] "Use of High Frequency Ionospheric Channel Simulators", CCIR (see [4]) Recommendation 520-1 (1978–82), 1982.

[73] Ralphs, J. D.; Sladen, F. M. E.. "An h.f. channel simulator using a new Rayleigh fading method", The Radio and Electronic Engineer, vol. 46, No. 12, p. 579–587, December 1976.

[74] Shimbo, O.; Celebiler, M. I.. "The Probability of Error Due to Intersymbol Interference and Gaussian Noise in Digital Communication Systems", IEEE Transactions on Communication Technology, vol. COM-19, No. 2, p. 113–119, April 1971.

[75] Saltzberg, B. R.. "Intersymbol Interference Error Bounds with Application to Ideal Bandlimited Signaling", IEEE Transactions on Information Theory, vol. IT-14, No. 4, p. 563–568, July 1968.

[76] Milewski, A.. "New Simple and Efficient Bounds on the Probability of Error in the Presence of Intersymbol Interference and Gaussian Noise", IEEE Transactions on Communications, vol. COM-25, No. 10, p. 1218–1222, October 1977.

[77] McLane, P. J.. "Lower Bounds for Finite Intersymbol Interference Error Rates", IEEE Transactions on Communications, vol. COM-22, p. 853–857, June 1974.

[78] Matthews, J. W.. "Sharp Error Bounds for Intersymbol Interference", IEEE Transactions on Information Theory, vol. IT-19, No. 4, p. 440–447, July 1973.

[79] Jenq, Y. C.; Liu, B.; Thomas, J. B.. "Probability of Error in PAM Systems with Intersymbol Interference and Additive Noise", IEEE Transactions on Information Theory, vol. IT-23, No. 5, p. 575–582, September 1977.

[80] Foschini, G. J.; Salz, J.. "Digital Communications Over Fading Radio Channels", Bell System Technical Journal, vol. 62, No. 2, p. 429–456, February 1983.

[81] Falconer, D. D.; Sheikh, A. U. H.; Eleftheriou, E.; Tobis, M.. "Comparison of DFE and MLSE Receiver Performance on HF Channels", IEEE Transactions on Communications, vol. COM-33, No. 5, p. 484–486, May 1985.

[82] Eleftheriou, E.; Falconer, D. D.. "Steady-State Behavior of RLS Adaptive Algorithms", ICASSP 85, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (Cat. No. 85CH2118-8), Tampa, FL, USA, 26–29 March 1985 (New York, USA; IEEE 1985), vol. 3, p. 1145–1148.

[83] Mueller, K. H.; Spaulding, D. A.. "Cyclic Equalization—A New Rapidly Converging Equalization Technique for Synchronous Data Communication", Bell System Technical Journal, vol. 54, No. 2, p. 369–406, February 1975.

[84] Clark, A. P.; Zhu, Z. C.; Joshi, J. K.. "Fast start-up channel estimation", IEE Proceedings, vol. 131, Pt. F, No. 4, p. 375–382, July 1984.

[85] Clark, A. P.; McVerry, F.. "Improved Channel Estimator for an HF Radio Link", Signal Processing, vol. 5, No. 3, p. 241–255, May 1983.

[86] Wilkinson, J. H.; Reinsch, C.. "Handbook for Automatic Computation", Volume II, Linear Algebra, p. 9–11, Springer-Verlag, 1971.

[87] Golub, G. H.; Van Loan, C. F.. "Matrix Computations", p. 125–135, North Oxford Academic, 1983.

[88] Clark, A. P.; Hau, S. F.. "Adaptive adjustment of receiver for distorted digital signals", IEE Proceedings, vol. 131, Pt. F, No. 5, p. 526–536, August 1984.

[89] Sato, Y.. "A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems", IEEE Transactions on Communications, vol. COM-23, p. 679–682, June 1975.

[90] Godard, D. N.. "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems", IEEE Transactions on Communications, vol. COM-28, No. 11, p. 1867–1875, November 1980.

[91] Benveniste, A.; Goursat, M.. "Blind Equalizers", IEEE Transactions on Communications, vol. COM-32, No. 8, p. 871–883, August 1984.

[92] Foschini, G. J.. "Equalizing Without Altering or Detecting Data", AT & T Technical Journal, vol. 64, No. 8, p. 1885–1911, October 1985.

[93] Cioffi, J. M.; Kailath, T.. "Fast, Recursive-Least-Squares Transversal Filters for Adaptive Filtering", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, No. 2, p. 304–337, April 1984.

[94] Clark, A. P.; Harun, R.. "Assessment of Kalman-filter channel estimators for an HF radio link", IEE Proceedings, vol. 133, Pt. F, No. 6, p. 513–521, October 1986.

[95] Eleftheriou, E.; Falconer, D. D.. "Tracking Properties and Steady-State Performance of RLS Adaptive Filter Algorithms", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-34, No. 5, p. 1097–1110, October 1986.

[96] Eleftheriou, E.; Falconer, D. D.. "Adaptive Equalization Techniques for HF Channels", IEEE Journal on Selected Areas in Communications, vol. SAC-5, No. 2, p. 238–247, February 1987.

[97] Clark, A. P.; Abdullah, S. N.. "Near-maximum-likelihood detectors for voiceband channels", IEE Proceedings, vol. 134, Pt. F, No. 3, p. 217–226, June 1987.

[98] Clark, A. P.; Abdullah, S. N.; Ameen, S. Y.. "A comparison of decision-feedback equalizers for a 9600 bit/s modem", Journal of the Institution of Electronic and Radio Engineers, vol. 58, No. 2, p. 74–83, March/April 1988.

[99] Schwartz, M.; Bennet, W. R.; Stein, S.. "Communication Systems and Techniques", New York: McGraw-Hill, 1966, Chapter 9.

[100] Bitzer, D. R.; Chester, D. A.; Ivers, R.; Stein, S.. "A Rake System for Tropospheric Scatter", IEEE Transactions on Communication Technology, vol. COM-14, p. 499–506, August 1966.

[101] Shafi, M.; Moore, D. J.. "Further Results on Adaptive Equalizer Improvements for 16 QAM and 64 QAM Digital Radio", IEEE Transactions on Communications, vol. COM-34, No. 1, p. 59–66, January 1986.

[102] Turin, G. L.; Clapp, F. D.; Johnston, T. L.; Fine, S. B.; Lavry, D.. "A Statistical Model of Urban Multipath Propagation", IEEE Transactions on Vehicular Technology, vol. VT-21, p. 1–9, February 1972.

[103] Turin, G. L.. "Introduction to Spread-Spectrum Anti multipath Techniques and their Application to Urban Digital Radio", Proceedings of the IEEE, vol. 68, No. 3, p. 328–353, March 1980.

[104] Hashemi, H.. "Simulation of the Urban Radio Propagation Channel", IEEE Transactions on Vehicular Technology, vol. VT-28, p. 213–224, August 1979.

[105] Clark, A. P.. "Digital Modulation for Cellular Radio Systems", Technical Note, Telecommunications (USA), vol. 20, No. 9, p. 46, 48, 50, September 1986.

[106] IEE Colloquium on "Multiple Access Techniques in Radio Systems", Electronics Division. Organised by Professional Group E8 (Radiocommunication Systems), Tuesday 7 October 1986, Digest No. 1986/95.

[107] Bajwa, A. S.; Parsons, J. D.. "Small area characterisation of UHF urban and suburban radio propagation", IEE Proceedings, vol. 129, Pt. F, No. 2, p. 102–109, 1982.

[108] Karim, R.. "Packet Communications on a Mobile Radio Channel", AT & T Technical Journal, vol. 65, Issue 3, p. 12–20, May/June 1986.