ADAPTIVE MULTIPLE SYMBOL DECISION FEEDBACK FOR NON-COHERENT DETECTION

NISHKAR BALAKRISHNA GOVENDER

Submitted in fulfilment of the academic requirements
for the degree of MSc Eng
in the School of Electrical, Electronic and Computer Engineering
at the University of KwaZulu-Natal, Durban, South Africa

17 July 2006

Supervisor: Dr. H. Xu
Co-supervisor: Prof. F. Takawira

Abstract

Non-coherent detection is a simple form of signal detection and demodulation for digital communications. The main drawback of this detection method is the performance penalty incurred, since the channel state information is not known at the receiver. Multiple symbol detection (MSD) is a technique employed to close the gap between coherent and non-coherent detection schemes.

Differentially encoded *M*-ary phase shift keying (DM-PSK) is the classic modulation technique that is favourable for non-coherent detection. The main drawback for standard differential detection (SDD) has been the error floor incurred for frequency flat fading channels. Recently a decision feedback differential detection (DFDD) scheme, which uses the concept of MSD was proposed and offered significant performance gain over the SDD in the mobile flat fading channel, almost eliminating the error floor.

This dissertation investigates multiple symbol decision feedback detection schemes, and proposes alternate adaptive strategies for non-coherent detection. An adaptive algorithm utilizing the numerically stable QR decomposition that does not require training symbols is proposed, named QR-DFDD. The QR-DFDD is modified to use a simpler QR decomposition method which incorporates sliding windows: QRSW-DFDD. This structure offers good tracking performance in flat fading conditions, while achieving near optimal DFDD performance.

A bit interleaved coded decision feedback differential demodulation (DFDM) scheme, which takes advantage of the decision feedback concept and iterative decoding, was introduced by Lampe in 2001. This low complexity iterative demodulator relied on accurate channel statistics for optimal performance. In this dissertation an alternate adaptive DFDM is introduced using the recursive least squares (RLS) algorithm. The alternate iterative decoding procedure makes use of the convergence properties of the

RLS algorithm that is more stable and achieves superior performance compared to the DFDM.

Preface

The research work presented in this dissertation was performed by Nishkar Balakrishna Govender, under the supervision of Dr. Hongjun Xu, at the University of KwaZulu-Natal's School of Electrical, Electronic and Computer Engineering, in the Centre of Radio Access Technologies. The work was partially sponsored by Telkom SA Ltd and Alcatel as part of the Centres of Excellence programme.

The entire dissertation, unless otherwise indicated, is the author's work and has not been submitted in part, or in whole, to any other universities for degree purposes.

Acknowledgments

I wish to thank my supervisor, Dr. Hongjun Xu, for his excellent supervision, support and guidance. I am extremely grateful for his willingness to set aside his personal time to assist me with whatever problem I encountered. The incessant discussions and valuable input was paramount to the success of this dissertation. I would also like to thank Prof. F. Takawira for his advice and opinions on my research.

To my father and brother, you both have been my pillars of support, through all the trying times. The encouragement and advice is greatly appreciated. I would also like to thank my brother for proof reading parts of this dissertation, and for always taking a keen interest in all of my endeavours. I am indebted to my aunts for their never-ending desire to keep me fed.

I am also grateful for the financial support received from Telkom SA. Ltd and Alcatel, without which my postgraduate studies would not have been the entertaining and fulfilling time it was.

To my friends and postgraduate colleagues, the bonds we have forged during the times of frustration and joy will last a lifetime. I thank you all for making my studies truly memorable.

Lastly to I wish to thank my mother. She was my compass, my friend, my healer, my protector, my teacher and much more. It is because of her, that I am who I am. I realize now that life is short, and we must try and live, and achieve the best we can.

Table of Contents

Abstract..........i

Preface..........iii

Acknowledgments..........iv

Table of Contents..........v

List of Figures..........ix

List of Tables..........xi

List of Acronyms..........xii

Mathematical Notations..........xv

Chapter 1 Introduction..........1

1.1 General..........1

1.2 Digital Communication Systems..........2

1.2.1 An Overview..........2

1.2.2 Channel Coding and Decoding..........3

1.2.3 Modulation..........5

1.2.4 Wireless Channels: Types and Models..........5

1.3 Motivation for Research..........7

1.4 Dissertation Overview..........9

1.5 Research Contributions of the Dissertation..........10

Chapter 2 Current Literature and Theory..........13

2.1 MSD Techniques for Improved Uncoded Differential Detection..........13

2.1.1 Multiple Symbol Differential Detection..........13

2.1.2 MSD over Flat Fading Channels..........14

2.1.3 MSDD with Decision Feedback 15

2.2 MSD Techniques for Coded Non-coherent Systems 16

2.2.1 Bit Interleaved Coded Modulation (BICM)...16

2.2.2 Non-coherent Sequence Detection...17

2.2.3 Iterative Decision Feedback Differential Demodulation 18

2.2.4 Iterative Non-coherent Detection with Soft Metrics.................................18

2.2.5 Coded MSD Techniques with Antenna Diversity 19

2.3 Adaptive Filter Theory...20

2.3.1 Adaptive Filters...21

2.3.2 Development of AFT 23

2.3.2.1 Statistical Approaches 23

2.3.2.2 Deterministic Approaches 24

2.3.2.3 Combinational Approach...24

2.3.3 Linear Prediction Filters 25

2.3.4 The Recursive Least Squares Algorithm 28

2.3.4.1 Summary of exponentially weighted RLS Algorithm...........................30

2.3.4.2 Convergence Properties of the RLS algorithm......................................31

2.3.5 The QR Factorization..31

2.3.5.1 Computation of QR Decomposition...33

2.3.5.2 The Givens Rotations 34

2.4 Adaptive Techniques for Non-coherent Detection 35

2.4.1 Adaptive Channel Estimation 36

2.4.2 Decision Feedback Adaptive Linear Prediction 36

2.4.3 DFDD with RLS 36

2.4.4 Adaptive Detection and Decoding via Mixture Kalman Filtering.............37

Chapter 3 Fundamental Decision Feedback Schemes 38

3.1 Decision Feedback Differential Detection (DFDD)..39

3.1.1 System model 39
3.1.2 Derivation of DFDD Metric 40
3.1.3 DFDD Detector 43

3.2 Adaptive DFDD 44

3.2.1 Prediction based DFDD 45
3.2.2 Adaptive DFDD with RLS algorithm 48
3.2.3 Equivalence of DFDD and Prediction based DFDD 50

3.3 Iterative Decision Feedback Differential Demodulation 51

3.3.1 System Model 52
3.3.2 DFDM Metric Derivation 53
3.3.3 The Iterative Decoding Algorithm 54
3.3.4 Convergence and Cut-off Rate Analysis 55

3.4 Simulation Results 59

3.4.1 Standard DFDD Results 59
3.4.2 RLS-DFDD Results 61
3.4.3 Iterative DFDM Results 63

3.5 Conclusion 64

Chapter 4 Adaptive DFDD with the QR Decomposition 65

4.1 The QR Decomposition and the Least Squares Problem 66

4.2 Adaptive DFDD with the QR Decomposition 70

4.3 Adaptive DFDD with QR Decomposition and Sliding Window 72

4.4 Initialization of the QR based Algorithms 74

4.5 Summary of QRSW-DFDD 75

4.6 Simulation Results 76

4.6.1 Effect of Sliding Window Size 76
4.6.2 Effect of Forgetting Factor 80
4.6.3 Time Selective Fading Conditions 82

4.7 Conclusion 84

Chapter 5 Adaptive DFDM with RLS Algorithm 87

5.1 Adaptive DFDM metric..88

5.2 The Iterative Decoding Procedure 90

5.3 Convergence of Adaptive DFDM 91

5.4 Alternate Approach to Adaptive DFDM 91

5.5 Simulation Parameters and Models..92

5.6 Simulation Results 93

5.7 Conclusion 99

Chapter 6 Conclusion 101

6.1 Conclusion of Dissertation 101

6.2 Future Work 103

Appendices 105

A: Rayleigh Fading Generation Method 105

Bibliography 106

List of Figures

Fig. 1.1 Block diagram of a digital communication system 2

Fig. 2.1 Block Diagram of BICM scheme 16

Fig. 2.2 A standard transversal forward linear predictor structure with fixed tap weights 26

Fig. 2.3 Forward error predictor showing relationship with linear predictor 27

Fig. 3.1 The block diagram of the DFDD detector. 39

Fig. 3.2 The DFDD detector structure. 43

Fig. 3.3 Block Diagram of iterative DFDM 52

Fig. 3.4 Standard DFDD results for F_dT=0.03, for N=2,3,4, optimal coefficients used.. 60

Fig. 3.5 Standard DFDD results for non-optimal filter coefficients, with F_dT=0.03. 61

Fig. 3.6 A comparison of the DFDD vs. the RLS-DFDD. Optimal coefficients were used. F_dT=0.03 62

Fig. 3.7 BER results of the DFDM for observation windows of N=2, 3, 5. Associated genie aided curves and perfect CSI case also shown 63

Fig. 4.1 Sliding window operation on the data matrix 73

Fig. 4.2 Plot of the QRSW-DFDD genie aided bounds. Sliding window sizes of 3 and 20, and observation window size N=3. DFDD genie aided bound and SDD also shown 77

Fig. 4.3 Plot of the QRSW-DFDD and the DFDD with decision feedback. Sliding window sizes of 3 and 20, and observation window size N=3. SDD also shown... 78

Fig. 4.4 Plot of the genie aided QRSW-DFDD bounds for sliding window sizes of 5 and 20. SDD and the genie aided DFDD are used as performance benchmarks. 79

Fig. 4.5 Plot of the QRSW-DFDD for sliding window sizes of 5 and 20, for N=5. The SDD and the DFDD is shown as performance bounds. 80

Fig. 4.6 Plot of BER vs. constant forgetting factor at 60dB, sliding window of 20. The dashed lines are for the genie aided cases. 81

Fig. 4.7 Plot of BER vs. Exponential forgetting factor at 60dB, sliding window of 20. The dashed lines are for the genie aided cases. 82

Fig. 4.8 Comparative plot of RLS-DFDD, QRSW-DFDD and DFDD with optimum metrics, for rapid flat fading channel, F_dT=0.08 and N=3 83

Fig. 4.9 Comparative plot of RLS-DFDD, QRSW-DFDD and DFDD with optimum metrics, for rapid flat fading channel, F_dT=0.08 and N=4 84

Fig. 5.1 Comparative genie aided BER curves for the adaptive DFDM and the DFDM.93

Fig. 5.2 Comparative BER curves for the adaptive DFDM and the DFDM with decision feedback after 5 and 4 iterations respectively. 95

Fig. 5.3 The filter coefficients reached over 100 frames of training for N=3 with F_dT=0.01 at 10dB. The optimal Wiener Coefficients are also shown 96

Fig. 5.4 Plot of BER convergence for adaptive DFDM, 4 iterations for N=3 (solid lines), while 5 iterations for N=5 (dashed lines). 98

Fig. 5.5 BER of adaptive DFDM with genie aided performance bounds. Coherent detection with perfect CSI is also shown 99

List of Tables

Table 1.1 Classification of Fading Channels 6

Table 4.1 Summary of Adaptive DFDD Algorithms 86

Table 5.1 Summary of Iterative Decoding Scheme for Adaptive DFDM 91

List of Acronyms

3G	Third Generation
ACE	Adaptive Channel Estimation
ACF	Autocorrelation Function
ACM	Autocorrelation Matrix
AFT	Adaptive Filter Theory
AR	Autoregressive
ARQ	Automatic Repeat Request
ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem (code)
BER	Bit Error Rate
BICM	Bit Interleaved Coded Modulation
CD	Coherent Detection
CDMA	Code Division Multiple Access
CORDIC	Coordinate Rotation Digital Computation
CPM	Continuous Phase Modulation
CSI	Channel State Information
DFDD	Decision Feedback Differential Detection
DFDM	Decision Feedback Differential Demodulation
DM-PSK	Differentially Encoded M-PSK
DSB-SC	Double Sideband Suppressed Carrier

EHF	Extremely High Frequency
EM	Electromagnetic
FEC	Forward Error Correcting Code
FEP	Forward Linear Error Predictor
FLP	Forward Linear Predictor
FPGA	Field Programmable Gate Array
FSK	Frequency shift Keying
HCM	Hybrid Coded Modulation
ISI	Inter-symbol Interference
LAN	Local Area Network
LMS	Least Mean Square
LOS	Line of Sight
LSL	Least Squares Lattice
MAN	Metropolitan Area Network
MAP	Maximum A Posteriori
ML	Maximum Likelihood
MLC	Multilevel Coding
M-PSK	M-ary Phase Shift Keying
MSD	Multiple Symbol Detection
MSDD	Multiple Symbol Differential Detection
MSK	Minimum Shift Keying
NCD	Non-coherent Detection
OFDM	Orthogonal Frequency Division Multiplexing

PLL	Phase Locked Loop
QAM	Quadrature Amplitude Modulation
QR-DFDD	QR based DFDD
QRD-LS	QR decomposition Least Squares
QRSW-DFDD	QR sliding window based DFDD
RLS	Recursive Least Squares
RLS-DFDD	RLS based DFDD
RLSL	Recursive Least Squares Lattice
RS	Reed Solomon
SDD	Standard Differential Detection
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
SOVA	Soft Output Viterbi Algorithm
TCM	Trellis Coded Modulation
VLSI	Very Large Scale Integration

Mathematical Notations

$E\{\square\}$	- the expectation
$[\square]^H$	- the hermitian transpose
$(\square)^*$	- the complex conjugate
$\|\square\|$	- the Euclidean norm
$\square$	- the set of all natural numbers 1, 2, 3, ...
$[\square]^T$	- the transpose
$\lvert\square\rvert$	- the absolute value
z^{-1}	- discrete time delay of one symbol period
$\mathrm{Re}\{\square\}$	- the real part of a complex number
$\lvert\square\rvert_{\det}$	- the determinant
J_0	- the zeroth order Bessel function of the first kind

The bold face symbols represent vectors, or matrices.

Chapter 1
Introduction

1.1 General

We are in the midst of an exciting and dynamic digital age. Technology today allows for broadband Internet access, high definition multimedia streaming, mobile audio and video streaming for motor vehicles, the home and cellular. Most of these applications are now being done wirelessly. At the core of all these technologies is digital communication.

The amount of research being done in developing wireless technologies and applications have stemmed from the expeditious growth of wireless users. Today the number of wireless users has exceeded the number of wireline users. Cellular telephony has spurred the development of new mobile data services other than voice over wireless channels.

Wireless communications today forms an integral part of modem day living, and is used in a wide variety of applications. Recent wireless data applications are exemplified by the IEEE 802.1x standards. These new wireless applications will probably be collectively standardized under the next generation (4G) cellular, which include personal local area networks (localized networks e.g. Bluetooth), wireless local area networks (LANs) and HiperLAN (IEEE 802.11 family), wireless metropolitan area network (MAN 802.16 also known as WiMax systems), satellite communications and a few others.

1.2 Digital Communication Systems

This section gives a generic overview of a digital communications system mostly summarized from [1]. Each element will be discussed briefly, while associated research developments are highlighted.

1.2.1 An Overview

The goal of any communication system is to ensure that all the data gets from the source to the required sink with no errors, and in as efficient a way as possible. A typical digital communication system is depicted in Fig. 1.1.

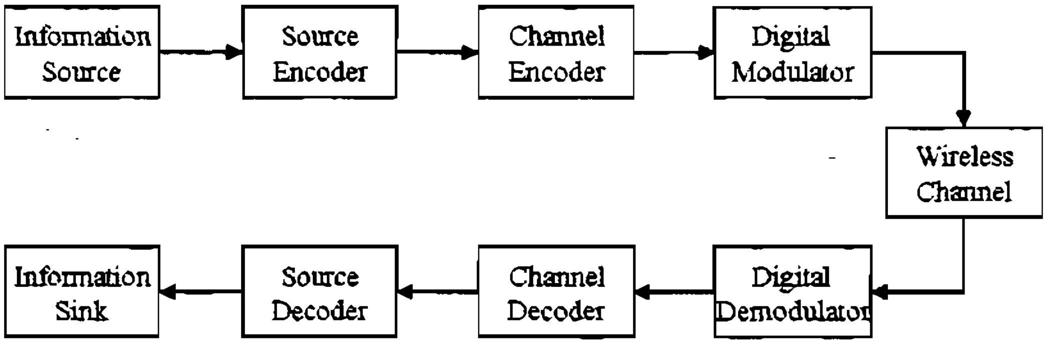

Fig. 1.1 Block diagram of a digital communication system

The information source may be digital or analogue in nature. The source information is then digitized or compressed into binary words, which are sequentially fed into the channel encoder. The channel encoder introduces a structured form of redundancy into the binary words so they may be readily and reliably decoded at the receiver. A further discussion of channel coding is warranted and will be covered later. The coded binary words (codewords) are now mapped to some finite constellation which will be used to transmit the information across the channel. The modulator converts the codewords into readily detected analogue signals. The choice of modulation depends on the channel characteristics and application.

The wireless channel, due to its physical constraints corrupts the channel signals in both amplitude and phase, which could result in incorrect detection. Wireless channels suffer primarily for three reasons namely: additive white Gaussian noise (AWGN), multipath fading, and interference.

The demodulator determines an estimate of the transmitted signal based on the received corrupted version and transfers the estimate of the codewords to the decoder. The channel decoder attempts to reconstruct the original information sequence, while correcting as many errors as possible.

A performance measure of the demodulator and decoder is the bit error rate (BER), where the decoded sequence of binary information is compared to the original sequence. The average occurrence of errors is equivalent to the probability of bit error.

The source decoder transforms the binary words into their original form (mostly for analogue outputs), knowing what source coding was employed. This is an estimate of the original analogue sequence, because due to the generally quantized nature of source encoding an exact copy is not possible.

1.2.2 Channel Coding and Decoding

Channel coding is usually performed using some forward error correcting codes (FEC). A bit sequence of length k is represented by a unique bit sequence of length n called a codeword. The amount of redundancy is measured using the ratio n/k, where $n>k$. The reciprocal of this ratio is known as the code rate.

There are many different types of coding schemes and this is still very much an active area of research. The encoding may be applied to blocks of data as in block codes, or sequentially as in convolutional codes. It is also important to mention the concatenated code, where multiple codes are used to encode the information. Typical codes encountered in practice are Hamming codes, Hadamard codes, Golay codes, cyclic Hamming codes, Bose-Chaudhuri-Hocquenghem (BCH) codes and Reed Solomon (RS) codes [2], which may be regarded as non-binary BCH codes. Many efficient algorithms exist to decipher these coding techniques, which make them favourable for implementation.

Convolutional codes, first proposed in [3], pass the information bits through a linear m-stage shift register, and at each instant in time there exists n encoded bits which are a function of the k bits, and the m-stage shift register. The optimal maximum likelihood (ML) decoding algorithm was introduced by Viterbi in [4]. Many telecommunication applications stemmed from this ML sequence detector, and the Viterbi decoder is still being used today. Non-trivial soft decision based algorithms like the soft output Viterbi algorithm (SOVA) and the maximum a posteriori (MAP) algorithm may also be used for convolutional codes.

Concatenated coding was introduced in 1966 by Forney [5], which combined an RS code with a convolutional code. Concatenated codes have already found applications in space communications with NASA's Galileo mission. A new class of concatenated codes emerged in 1993 by Berrou, Glavieux and Thitimajshima [6], called "Turbo codes". Turbo codes performed within 1 dB of the Shannon limit in AWGN channels. The main contribution of this type of code was not just the performance but the algorithm itself, which could be adopted in other ways. Berrou et al. introduced two recursive systematic convolutional codes in parallel, separated by a pseudo random interleaver. The decoding process was iterative with extrinsic information being exchanged, thereby decreasing the number of errors after each iteration. The decoding process itself has been applied to numerous other problems, including serial concatenation [7, 8], channel equalization [9], CDMA systems [10, 11], automatic repeat request (ARQ) protocols for error control [12], joint source channel decoding [13], and non-coherent differential detection [14, 15]. The concept of Turbo codes was indeed a significant contribution in the area of channel coding.

1.2.3 Modulation

There are a host of digital modulation techniques that may be used to map the encoded binary words to some analogue form. The selection of a specific modulation technique is dependent on the channel characteristics, the intended application and the susceptibility of the channel to fading. A number of digital modulation techniques are described in [16] e.g. M-ary phase shift keying (M-PSK), frequency shift keying (FSK), amplitude shift keying (ASK), etc. In this dissertation, the differential M-PSK type of modulation is used, where non-coherent methods are employed to decipher the binary words.

1.2.4 Wireless Channels: Types and Models

Mobile telecommunication occurs over wireless channels, therefore it is important to understand the properties and limitations placed on wireless channels. The requirements for wireless communication are based on the frequency of operation, especially the size and configuration of the antenna. The frequency bands of use currently range from the low frequency audio bands to the extremely high frequency (EHF) band.

The problem with electromagnetic (EM) propagation (apart from noise, which is ubiquitous for all frequencies) in the higher frequency bands is signal multipath propagation. The transmitted signal arrives at the receiver, after taking several paths due to reflection or refraction, at different time delays. These different signals could interfere constructively or destructively with each other. This multiple signal effect is referred to as multipath fading. It severely inhibits reliable detection due to its random nature. Another phenomenon experienced in mobile communications related to this multipath fading is the Doppler shift associated with the velocity of the mobile terminal. Considering these effects a mathematical model is developed to model the concept of fading.

Fading is treated in terms of the time and frequency domains. In terms of the multipath effect the largest delay among the various multipaths is referred to as the delay spread (τ_d). The reciprocal of the delay spread is defined as the coherence bandwidth

(B_d □ $1/\tau_d$). Using the coherent bandwidth it can be determined whether the channel would be frequency selective or frequency non-selective.

The Doppler spread (F_d), which is the maximum Doppler frequency shift associated with the channel, due to the continuous motion between the transmitter and receiver, is the mobility parameter. The time domain counterpart is the coherent time, defined as T_d □ $1/F_d$, the reciprocal pf the Doppler spread. These parameters are used as an indicator for time selectivity.

Letting the bandwidth of the transmitted signal be denoted by B_S, and the period of the channel symbol T_S and using the definitions of the coherent bandwidth and coherent time, channels may be classified. The classification of the fading channel is described in Table 1.1, summarized from [17].

Table 1.1 Classification of Fading Channels

	B_S □ B_d	$B_S \geq B_d$
T_S □ T_d	Frequency non-selective Time non-selective	Frequency selective Time non-selective
$T_S \geq T_d$	Frequency non-selective Time selective	Frequency selective Time selective

The term frequency selective arises from the fact that under these channel conditions the fading gains are different for different frequencies, while a similar argument applies to time selectivity. It is worth mentioning another case where there is no Doppler shift. The channel remains virtually static for the entire frame of data to be transmitted, this case is referred to as the quasi-static fading channel.

Statistical models are used to represent the classified channels in Table 1.1, because these multipaths are random in nature. Due to central limit theorem the amplitude of the fading may be modelled as a complex Gaussian random variable, with the envelope having different statistical distributions. Typical distributions include Rayleigh fading, where the multipaths are scattered and no dominant line of sight (LOS) path exists between transmitter and receiver, or Ricean fading where a dominant LOS path exists and the scattered components contribute little to fading amplitude.

1.3 Motivation for Research

Information in most digital communication systems today are conveyed through the modulation of the phase of the carrier wave, such as M-PSK. There are primarily two ways of detecting these signals: coherent detection or non-coherent detection.

Ideally the symbol transmitted may be determined just by observing the phase of the carrier during a particular time interval. The signal is first demodulated, through the use of a local oscillator, and the resulting signal is then filtered to adequately nullify the effects of noise. A phase discriminator is then applied to determine the resulting symbol. Typically, there is a carrier phase offset resulting from the channel path and the time taken to traverse the wireless channel. To accurately demodulate the M-PSK symbols, the phase offset has to be known or estimated with great certainty. Coherent detection has accurate knowledge of the phase offset which is removed during demodulation, while non-coherent detection removes the phase offset after demodulation.

When using coherent detection, the main problem experienced in practical systems is synchronization. The receiver has to be synchronous for both the time propagation delay (symbol synchronization) between the transmitter and receiver, and the phase offset (carrier synchronization). Mathematically the phase offset may be estimated using the propagation delay time. However, the problem of oscillator drift at both transmitter and receiver does not make this reliable. There are two basic approaches to carrier phase estimation at the receiver. One approach is to use a pilot tone in addition to the information carrying signal. A phase-locked loop (PLL) is then employed at the receiver to acquire and track this phase. The second approach which is more efficient in terms of the power and bandwidth is to derive the carrier phase directly from the modulated carrier. This sort of transmission is known as a suppressed carrier system, e.g. double sideband suppressed carrier (DSB-SC). The latter approach is the one most prevalent in modern communication systems. There are however problems with both methods. Detailed methods of carrier and symbol synchronization are discussed in [1] and [18].

In the pilot approach, the acquisition time may be long, especially for mobile communications. The PLL may also encounter false locks or lock slips in noisy environments. This will degrade the response and performance of the coherent detector. The second method when generalized for *M*-ary constellations suffers from multiple phase ambiguities, and high complexity for large constellations. The overall analogue circuitry required for both methods may be too taxing for the receiver, therefore non-coherent detection is the natural alternative. This neither requires knowledge of the carrier nor the complex circuitry involved for carrier and symbol synchronization.

Non-coherent detection is generally employed for differentially encoded *M*-PSK (DM-PSK), where the information is carried between the phases of sequential channel symbols. The alternative non-coherent strategy, uses absolutely encoded *M*-PSK signals and pilot symbols to estimate the phase. This method is not efficient in terms of information capacity and transmission power. DM-PSK is therefore still the most common form of digital modulation used in non-coherent detection.

Mobile channels are typically slow fading environments where speeds of about 100km/h are the norm, however, third generation (3G) European standards for example, are required to operate on trains travelling up to 500km/h [19]. At this speed large Doppler shifts are expected and so the channel may be regarded as fast fading. In this case differential encoding in conjunction with channel coding has to be utilized for realizable performance. In fading channels the use of interleaving is essential, which in conjunction with convolutional coding offers protection against fading and noise. The use of an interleaver allows for an iterative decoding structure and mitigates the effect of fading.

Multiple symbol detection (MSD) first proposed in [20] offered a solution to try and bridge the gap between coherent and non-coherent detection. The introduction of MSD marginally increased the complexity of the non-coherent receiver, but improved the performance. The concept of MSD was to use multiple symbols to more accurately determine the phase error. MSD applied to non-coherent detection is still an active area of research as will be shown in the literature covered in chapter 2. Non-coherent receivers still do not match the performance of the coherent schemes, but the performance gap is closing. MSD techniques are beginning to shrink the performance gap which is often called the non-coherence penalty.

Conventional differential detection encounters a performance floor in flat fading environments. Many MSD techniques require the channel statistics for optimal detection in fading environments, so there is still some dependence on the channel. In mobile communications, and typically fast flat fading channels, the statistics measured and calculated may be inaccurate and unreliable. Therefore adaptive filtering forms an attractive solution in making these MSD schemes totally independent of the channel. Adaptive schemes and MSD based non-coherent detection offer viable solutions for use in mobile communication.

This dissertation focuses on adaptive non-coherent techniques for flat fading channels, in conjunction with MSD. The MSD technique used is centred on a simple decision feedback scheme which is used in systems with and without channel coding. Adaptive low complexity iterative decoding is also investigated.

1.4 Dissertation Overview

This dissertation is divided into six chapters. Chapter two covers the literature regarding the important non-coherent MSD detection schemes and the fundamental theory prevalent in the research covered. The relevant algorithms and structures of the MSD schemes are described. Standard linear filter theory and related adaptive filter techniques are also summarized.

In chapter three the prominent MSD decision feedback differential detection (DFDD) scheme and its adaptive counterpart are reviewed and their respective results replicated. A brief explanation of the iterative DFDM scheme for flat fading channels follows, with the associated BER performance shown. These existing schemes form the framework for the new research that is covered in the ensuing chapters.

The new alternate adaptive DFDD scheme using the QR decomposition is introduced in chapter four. The QR least squares solution and the resulting QR-DFDD scheme is derived. The modified QRSW-DFDD scheme is detailed and compared to the existing DFDD schemes. Simulation results comparing these DFDD schemes are shown for time non-selective and selective, flat fading channels.

A new adaptive DFDM scheme and its associated iterative process are discussed in chapter five. The metrics for the new adaptive DFDM algorithm and the application of the RLS algorithm is described. The convergence and number of iterations of the new adaptive structure are discussed. A performance comparison between the existing DFDM and the new adaptive DFDM structure is shown through BER simulation curves.

Chapter six draws the relevant conclusions from the dissertation, and discusses possible future work.

1.5 Research Contributions of the Dissertation

The DFDD detector [21], is a novel, simple MSD detector using decision feedback symbols to improve the performance of non-coherent differential detection. The DFDD has a linear predictor structure, which is advantageous for the application of adaptive filtering.

An alternate DFDD approach was introduced by the same authors in [22], which includes an adaptive scheme utilizing the RLS algorithm, referred to in this dissertation as RLS-DFDD. All these systems have similar decision feedback and linear filter structures. The mathematical equivalence between the DFDD of [21], and the linear prediction based DFDD of [22], for Rayleigh fading channels is shown in chapter 3 of this dissertation.

An alternate adaptive scheme is proposed using decision feedback in chapter 4. This adaptive decision feedback scheme utilizes the highly modular, pipelined and numerically stable QR decomposition, which is called the QR-DFDD algorithm. An alternate simplified approach to the QR-DFDD is also proposed using sliding windows, referred to as QRSW-DFDD. Both QR decomposition methods do not require any channel information and the adaptive process is based solely on the received sequence, so the adaptation is blind. The DFDD and its associated adaptive schemes are simulated for both time non-selective and time selective environments. The simulation results show that QRSW-DFDD is superior to RLS-DFDD in fast fading conditions, while maintaining a comparable performance to the DFDD and RLS-DFDD in flat fading environments with less mobility.

An iterative non-coherent scheme using channel coding was introduced in [23], which utilized the DFDD metric. This scheme used the simple bit branch metrics, of the Viterbi decoder, and achieved high performance gains over conventional bit interleaved non-coherent detection. This scheme was called iterative decision feedback differential demodulation (DFDM), and used decision feedback symbols through the iterative process to improve the performance. In chapter 5, an adaptive scheme utilizing the RLS algorithm is introduced to eliminate the dependence of the DFDM on the channel statistics. The new iterative decoding procedure, with training, is introduced. The adaptive algorithm does not require pilot symbols therefore the training procedure is blind. The RLS algorithm is not used for continuous adaptation, therefore the complexity of this DFDM architecture is only marginally increased, while still maintaining the advantages of the DFDM. The performance of the adaptive DFDM was superior to that of the DFDM in flat fading conditions. The adaptive DFDM is stable at low signal-to-noise ratios (SNRs), and has improved performance at higher SNRs, confirmed with simulation results.

The original contributions of this dissertation are summarized below:

1. The mathematical equivalence of the DFDD and linear prediction DFDD is derived for Rayleigh fading channels. The literature mentions that the schemes are equivalent, but it has not been shown mathematically.

2. An alternate adaptive scheme for the DFDD scheme is proposed using the QR decomposition called QR-DFDD. A simpler QRSW-DFDD is also proposed using a sliding window approach. Both QR based DFDD structures are favourable to VLSI. The QR based algorithms offer stable performance in fast fading environments, and comparable performance otherwise.

3. A new approach to the iterative DFDM scheme using prediction based metrics and the RLS algorithm is introduced. The new iterative procedure involves a blind training iteration. The new adaptive DFDM is independent of the channel state and the channel statistics, with superior performance when compared to the DFDM.

Chapter 2

Current Literature and Theory

The research documented in this dissertation covers various aspects of digital communications, digital signal processing as well as adaptive filtering. In recent years the area of non-coherent detection has been receiving much attention. Some of the notable literature will be covered in this chapter, while additional theory used in this dissertation will also be covered.

The most notable improvement of non-coherent detection (NCD) has been through the use of multiple symbol observation windows. The use of multiple symbol detection (MSD) has been applied to many different types of uncoded and coded systems.

2.1 MSD Techniques for Improved Uncoded Differential Detection

The concept of MSD is not a new one; it was documented in one form or another since 1988 [20, 24, 25]. This section describes notable improvements for uncoded systems using non-coherent detection and MSD.

2.1.1 Multiple Symbol Differential Detection

The concept of MSD was first applied to communications systems by Divsalar and Simon in [20]. In general for an additive white Gaussian noise (AWGN) channel, coherent detection (CD) of M-PSK symbols outperforms differential detection by approximately 3dB, even for coded systems. The authors in [20] introduced a maximum likelihood (ML) metric to determine the most likely sequence of symbols based on an observation window of size N. It was seen that this method of multiple symbol differential detection (MSDD) held great promise in bridging the performance gap between CD and standard differential detection (SDD).

As a special case SDD was equivalent to the MSDD with $N = 2$, where an observation window of size two is required, just as in SDD where the decision is made on a symbol-by-symbol basis. It was shown for multiple constellation sets that the MSDD metric offers significant performance gain with just finite window sizes e.g. $N = 3,5$, while the ideal coherent detection is approached if $N \to \infty$. However, it should be mentioned that the complexity for an M-ary signal is exponential with N where the number of trial sequences is M^{N-1}. This would mean that for a window $N = 5$, for constellations of $M = 4,8,16$, the corresponding number of trial sequences would be 256, 4096 and 65536 respectively. A method for decreasing this complexity was introduced in [26], where a smaller set of samples are used and where the complexity was decreased. The reduction factor varied from 16 to 4096 due to constellation size.

2.1.2 MSD over Flat Fading Channels

The performance of MSDD virtually bridged the gap between CD and NCD for AWGN channels. The performance for the correlated Rayleigh flat fading channel was investigated in [27], where it was clearly seen that SDD suffered an error floor for fading channels with large Doppler frequencies. The performance of this MSD system in the presence of fading eliminated the error floor by choosing the ML sequence of symbols. For an AWGN channel the decision metric became exactly that of MSDD in [20], while for the fading channel the decision metric included the channel fading characteristics. This system however, suffered the same problem as MSDD where for large observation windows the number of phase sequences to be searched became exponentially large.

An alternate method of MSDD using trial sequences was introduced in [28] which is suboptimal to [27], but achieves near identical performance. In [28], an alternate expression for the fading process in terms of the Karhunen-Loève expansion is derived, which is used to formulate the new decision metric. This detection algorithm also requirés a large number of trial sequences and has the same complexity as [27].

Another method of MSD using sphere decoding was introduced in [29]. This uses ordered trial sequences, therefore it performed better than the decision feedback differential detection (DFDD) scheme discussed in the next section, but at increased complexity. However, the ordered sequences become more reliable at higher SNRs and hence not all trial sequences are used, and in this case the complexity was shown to be less than the DFDD.

2.1.3 MSDD with Decision Feedback

The previous MSD schemes have one common detriment; the number of phase search sequences. An alternate method to trial sequences, was introduced by Edbauer in [25]. This is a method of using previously deciphered symbols and using them heuristically to determine the current ML symbol. It showed performance gains over SDD, and at high SNRs, approached the case where correct symbols were fed back, which will be referred to as the genie aided case. However, this feedback scheme was not generalized to flat fading channels.

In [21], Schober introduced the optimal decision feedback detection scheme for flat fading channels referred to as decision feedback differential detection (DFDD). This detection scheme offered low complexity metrics, and virtually eliminated the error floor always experienced in frequency non-selective fading channels.

In [22], the same author of the DFDD introduced an alternate DFDD structure using linear prediction. This structure was focused for use in Ricean fading channels, but could still be used for Rayleigh fading channels. In the same paper, an adaptive DFDD scheme, using the RLS algorithm, was introduced which performed just as well as the DFDD in flat Rayleigh fading channels.

These DFDD schemes are covered in greater detail in a chapter 3, because they provide the foundation of the research introduced in this dissertation. When compared to other MSD schemes the DFDD offered better performance with reduced complexity in most cases.

The DFDD concept was further extended to multi-antenna systems. The uncoded case was introduced in [30], where its application to orthogonal frequency division multiplexing (OFDM) systems was also discussed. It should be noted that this dissertation focused on single antenna systems.

2.2 MSD Techniques for Coded Non-coherent Systems

The MSD techniques of the previous section are impractical for application in the real world because of the power constraints for reliable communication in a mobile fading environment. As mentioned earlier the performance between CD and NCD schemes for AWGN channels is approximately 3dB. In multipath fading channels this gap is significantly larger, since the CD knows the channel information.

In this section detection schemes using the MSD principles in conjunction with channel coding are outlined. Non-iterative schemes are introduced and then a low complexity iterative decoding algorithm is discussed. Alternative iterative decoding algorithms with high complexity are subsequently described.

2.2.1 Bit Interleaved Coded Modulation (BICM)

A traditional BICM system is depicted in Fig. 2.1, which is applicable for both coherent and non-coherent schemes. For a non-coherent system the modulator will be appended by a differential encoder. The key feature in the structure is that there is separate demodulation and decoding, which is favourable in Rayleigh fading channels.

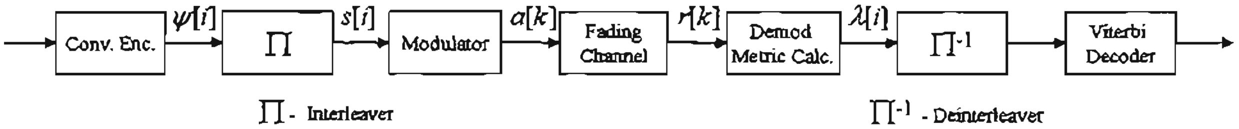

Fig. 2.1 Block Diagram of BICM scheme

A standard off the shelf convolution code (A table of standard convolutional codes can be found in [1]) and bit interleaver are standard features in the transmitter. The interleaving is assumed to be ideal, however, it was shown that practically, interleavers with finite shallow depth performed almost as well. The interleaver was used to nullify the relationship between encoder output and decoder input, essentially making the channel memoryless. The decoder employed a soft metric computation ($p(\mathbf{r}_k \mid \mathbf{a}_k)$) based on the modulation scheme, and deinterleaved the metrics as input to the standard Viterbi decoder. As can be seen a standard BICM system has a non-iterative structure with no feedback.

Although the importance of labelling was mentioned in [31], a more thorough investigation was made in [32], where the traditional BICM system was converted into an iterative system with hard decision feedback. It was shown in [32], that the iterative scheme performed better than standard trellis coded modulation (TCM) in fading channels. The structure was very similar to that proposed in [33], however for [32], the scheme was coherent and assumed accurate knowledge of the channel state information. BICM was generalized in [31], and offered an analytic framework for BICM systems. BICM did not include any MSD technique, but this system formed the backbone of the DFDM, which is discussed later.

2.2.2 Non-coherent Sequence Detection

In [34], the authors exploited the fact that MSD was known to improve the performance of non-coherent detection. The authors introduced a shortened trellis where each trellis state was defined in terms of the truncated trial information sequence. This scheme was also generalized for the differential encoding structure for quadrature amplitude modulation (QAM). The analytic bounds for coded PSK were given in [35]. The analysis and results were applicable to channels with noise, and inter-symbol interference (ISI).

It should be noted that this scheme is not iterative, and has complexity issues for large constellations or large observation windows, which was referred to as implicit phase memory in [34]. An additional deficiency was the absence of interleaving, which for flat fading channels is a necessity.

2.2.3 Iterative Decision Feedback Differential Demodulation

Using the BICM principle of keeping the demodulator and channel decoder separate, a feedback scheme utilizing hard decisions from the Viterbi decoder is fed back to the demodulator. This feedback scheme known as iterative decision feedback differential demodulation (DFDM) was introduced in [33]. The demodulator used the simple DFDD metric to exploit performance gains of MSD and enhances the performance after ensuing iterations. It was shown in [23] that with larger observation windows the performance gap between the CD and conventional NCD was decreased significantly.

The conventional NCD curve was equivalent to BICM with SDD applied, and was also equivalent to a special case of the DFDM when the observation window was limited to two. The DFDM system was recently modified by the original authors to improve the performance in [36], which included a more in depth analysis of the DFDM scheme. This new scheme is now also able to work with alternate labelling structures, unlike the DFDM which only showed performance improvement for Gray labelling.

It should be mentioned that this scheme is suboptimal, since there exists other strategies to improve the performance using soft decision feedback, but these are also more complex. A major contribution of this scheme was the trade-off between improved performance and low complexity. Alternate iterative structures with soft metrics are discussed in the next section. It should be noted that the original DFDM scheme is about 2dB worse than [15], where virtually the same conditions existed. However, the complexity, in contrast to the DFDM is extremely high.

2.2.4 Iterative Non-coherent Detection with Soft Metrics

There exist multiple iterative schemes that employ soft metric feedback for NCD. One such method was introduced in [14], which was essentially equivalent to the turbo coding scheme of [15]. In this system a trellis structure was used to represent the demodulation metric where the differential encoding process was regarded as a very high rate code. In addition a linear predictor was used to estimate the fading process, while the maximum *a posteriori* (MAP) algorithm was used to decode the ML symbols.

This yielded significant improvement over coded SDD, while improving on coherent detection by 0.2dB after three iterations. This turbo decoding scheme utilized the soft-input soft-output (SISO) structure, where soft metrics were passed to the channel decoder and soft metrics were fed back and extrinsic information extracted.

An alternate turbo decoding scheme was proposed in [37], which offered two versions of turbo decoding techniques. One version maximized the probability of the correct symbol detection using the Bahl et al. (BCJR) algorithm, a form of the MAP algorithm. The other version maximized the probability of choosing the most likely sequence of information through the use of the soft output Viterbi algorithm (SOVA). However, these performances may not be used for comparison with the other iterative schemes since the results published were only for an AWGN channel, while fading channels were used otherwise.

Based on the application of sphere decoding to differential detection, an alternate iterative turbo scheme with SISO decoding was introduced in [38] for flat fading channels. It utilized the MAP algorithm and near channel capacity results are achieved.

2.2.5 Coded MSD Techniques with Antenna Diversity

Antenna diversity was combined with MSD for uncoded systems in [30]. However, two notable coded schemes applied MSD to exploit the spatial diversity offered by multi-antennae systems. In [39], an alternate form of the DFDD metric was used to create a multi-antenna iterative DFDM structure. It is also important to note that orthogonal constellations are favourable to space time modulation. This scheme was applied to Ricean channels, since the original DFDD metrics of [21], was derived for Rayleigh channels. The alternate DFDD metric was based on the metric in [22], which used linear prediction coefficients.

Another coded scheme that exploited both multiple antennae and MSD was presented in [40]. However, this system was not iterative. It compared different channel coding schemes, and extended the BICM system to multiple antennae. Other coding structures explored were multilevel coding (MLC) and hybrid coded modulation (HCM) which is a combination of BICM and MLC.

Note this section introduced multi-antenna systems, but this dissertation focused on the single antenna case. It is possible that the algorithms presented in this dissertation may be employed in multi-antenna systems.

This concludes the summary of pertinent literature on MSD techniques. There are alternate schemes using linear predictors and adaptive linear prediction, these will be summarized after some basic filter theory is introduced in the next section.

2.3 Adaptive Filter Theory

Adaptive filter theory (AFT) is still an active research area in its own right. However, since aspects of AFT are employed in this dissertation an endeavour is made to introduce the topic as well as algorithms used.

From an information theoretic point of view, a filter is a device (structure) or algorithm that is used to extract useful or specific information from a set of received and available data. Filters can be used to perform three basic information processing operations, namely: filtering, smoothing and prediction.

A digital signal processing perspective is used to define these processes. Filtering is the extraction of information for a required quantity at time t by using data determined or measured up to and including time t. Smoothing is similar to filtering, but differs in the fact that it can make a decision on the quantity of interest later than time t, and it can incorporate data captured after time t. Prediction is a forecast of the quantity of interest using data up to and including time t, for some time $t+\tau$, where $\tau > 0$. These are based on the definitions in [41].

In general these filter structures are linear; meaning the filter is a linear function of the measured or the observed data received as inputs to the filter. Filter design may be approached from a statistical point of view where certain statistical characteristics of the input are known, and hence the filter structure would be fixed accordingly. However, there may be occasions where the required statistics are unavailable, therefore an adaptive filter offers an attractive solution to the filtering problem. In many cases the adaptive versions may actually offer performance improvement over their fixed coefficient counterparts.

2.3.1 Adaptive Filters

In general, if an environment is stationary, the required statistics are available and an optimum filter may be designed by minimizing the mean square error. The resulting solution is often referred to as the Wiener solution and is optimal in the mean square sense.

An adaptive filter is a self designing filter that is continually changing to try and optimize the output. It would generally employ a recursive algorithm to account for the changes in the input and change the filter accordingly. In this fashion it will work satisfactorily when there is no knowledge of the relevant input characteristics available.

In a stationary environment the recursive algorithm after a finite number of iterations would approach the optimal Wiener solution. A non-stationary environment has statistics that are time variant hence fixed coefficients would be suboptimal, in this case the adaptive solution would offer a tracking capability. The statistical changes must be sufficiently slow, otherwise the adaptive filter will not work effectively.

Invariably the structure of a filter would be linear, adaptive or fixed. It is worth mentioning that although this structure is linear an adaptive filter is data dependent, and hence it may be regarded as non-linear. However, we will consider it from the structural approach and regard it as linear.

Numerous recursive algorithms have been developed and applied in an adaptive filter. However, each algorithm offers a trade-off between many factors. These performance criteria are outlined as in [41],

- Rate of convergence: the number of iterations before the adaptive solution reaches the optimal Wiener solution. This is important if the requirement is fast adaptation to an unknown environment, and also for tracking statistical variations in a non-stationary environment.
- Misadjustment (estimation error difference): A quantitative yardstick, the value by which the mean square error averaged over an ensemble of filters, deviates from the minimum mean square error that is produced by the Wiener Filter solution. If this value is small then that implies the adaptive process is working well, since it is close to the Wiener solution.
- Robustness: the ability of the algorithm to work satisfactorily, given the fact that the input data is very badly conditioned (i.e. the input suffers from excessive noise, and variations are too fast to track easily).
- Computational requirements: practical implications of the algorithm, namely the number of mathematical operations for each iteration indicating complexity, memory required to store data for each iteration, investment required to program the algorithm on a PC or some other form of computer, e.g., Field programmable gate array (FPGA), microprocessor, digital signal processor, etc.
- Structure: the information flow of the algorithm, determining the manner in which the algorithm may be implemented, e.g., an algorithm whose structure has high modularity, parallelism or concurrency is well suited for very large scale integration (VLSI) implementation.
- Numerical properties: numerical stability is important since some algorithms do not cope well with round off noise and the introduction of these errors affects future iterations. Some algorithms are not well suited for continuous adaptation, unless some rescue methods are employed. Another aspect which could affect the algorithm is the eigenvalue spread, or if the input data matrix is underdetermined.

Adaptive filtering has found applications in many vistas of science and engineering. The ability to operate effectively in an unknown environment and also track variations of input statistics are sought-after traits. Digital communications, image processing, acoustics, seismology, control systems, radar and sonar systems, etc. are some of the areas where adaptive filtering has been employed.

2.3.2 Development of AFT

The development of the recursive algorithms to update the adaptive filter was historically developed using three approaches, the statistic approach, the deterministic approach or a combination of the two. In [41], Haykin categorized these approaches slightly differently; the approaches were based on the traditional algorithms they were derived from. It also partially stemmed from the chronological order of the development of these algorithms.

2.3.2.1 Statistical Approaches

The solution to the linear filter problem based on the statistics of the inputs was determined independently by Kolmogorov, Krein and Wiener. Kolmogorov used the discrete time approach while Wiener used the continuous time approach. It was Wiener who derived the explicit formula for determining the optimum predictor for a process corrupted by noise, which was known as the Wiener-Hopf equation. The discrete time matrix form was later formulated by Levinson, which is now referred to as the normal equation

$$\mathbf{R}\mathbf{w}_0 = \mathbf{p}, \tag{2.1}$$

because it is so often used in linear filter theory. $\mathbf{R}$, $\mathbf{w}_0$ and $\mathbf{p}$ are the correlation matrix of the tap inputs, the tap weight vector of the optimum Wiener filter and cross correlation vector between the tap inputs and the desired response, respectively.

The adaptive filter would normally have a tapped delay line or transversal filter structure. The mean square error (the mean square value of the difference between the desired response and the filter output) is precisely a second order function of the tap weights. Hence the error surface may be viewed as a multidimensional paraboloid with a uniquely defined minimum point which corresponds to the optimum Wiener solution. The resulting algorithm from this approach was the least mean squares (LMS) algorithm, which was derived through manipulation of the normal equation and the method of steepest descent.

The LMS algorithm is computationally simple. However, common drawbacks are slow convergence, and sensitivity to the eigenvalue spread of the correlation matrix of the tap inputs. Hence the LMS algorithm would work effectively only in certain cases.

2.3.2.2 Deterministic Approaches

Unlike the Wiener solution approach, which uses a statistical approach to minimize the mean square error, these approaches minimize the sum of the weighted error squares, where the error is the difference between the desired response and output of the filter.

Among the deterministic approaches are the recursive least squares (RLS), the least squares lattice algorithm (LSL) and the QR decomposition least squares (QRD-LS) approach, etc. These approaches improve on the LMS algorithm; they are rapidly convergent and are virtually insensitive to eigenvalue spread. The RLS approach is computationally complex, while the LSL and the QRD-LS are highly pipelined and modular which make them favourable to VLSI manufacturing.

2.3.2.3 Combinational Approach

The Kalman filter approach is the typical case in point. The Kalman filter is defined by two equations, the plant equation and the measurement equation. The plant equation describes the dynamics of the system in terms of the state vector, while the measurement equation describes the measurement errors experienced by the system. Here again a transversal filter structure is used.

However, in the Kalman filter model a state describing optimal conditions is required for the Kalman filter to track, which may affect the rate of convergence. The Kalman filter offers superior performance to the LMS algorithm, and is insensitive to eigenvalue spread. The main drawback of the Kalman filter is the computational complexity.

It is also worth mentioning that for an estimation of a constant state, the RLS filter is equivalent to the Kalman filter, so it can be said that the RLS algorithm is the deterministic counterpart of the Kalman filter algorithm.

2.3.3 Linear Prediction Filters

One of the basic information processing operations is prediction. This section will discuss special cases thereof: the forward linear predictor (FLP) and the forward linear error predictor (FEP).The derivations and definitions are summarized from [41]. The FLP makes a prediction using the input samples $\mathbf{u}_L$,

$$\mathbf{u}_L(i-1) = \{u[i-1], u[i-2], \ldots, u[i-L]\}^T, \tag{2.2}$$

where i is the time index, and L is the effective length of the required input vector, also representing the order of the filter. It is assumed that the samples are drawn from a stationary stochastic process with zero mean. The prediction is a linear combination of the samples, therefore the prediction relation may be written as

$$\hat{u}[i] = \mathbf{p}\mathbf{u}_L(i) = \sum_{k=1}^{L} p_k u[i-k], \tag{2.3}$$

where $\mathbf{p}$ is the constant vector of tap weights defined as

$$\mathbf{p} \square [p_1, p_2, \ldots p_L]. \tag{2.4}$$

The corresponding transversal structure is shown in Fig. 2.2, where the order is determined by the number of unit time delays, which in this case is L.

The desired response $d[i]$ is equal to the input $u[i]$, hence the FLP error is the difference between the prediction and desired response written as

$$e_{FLP}[i] = d[i] - \hat{u}[i] = u[i] - \hat{u}[i]. \tag{2.5}$$

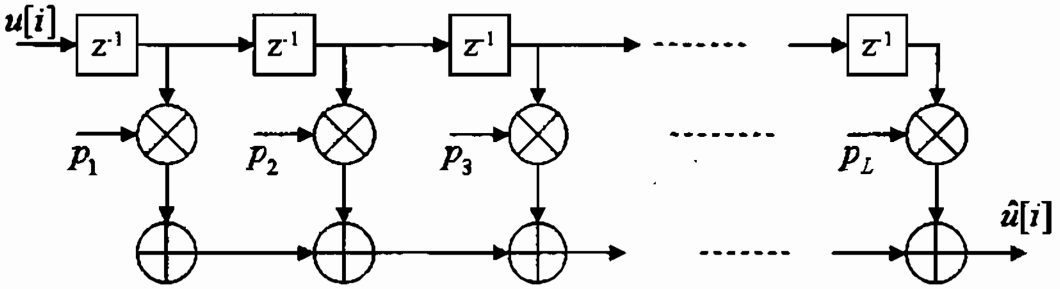

Fig. 2.2 A standard transversal forward linear predictor structure with fixed tap weights

Another quantity is introduced, P_e, which denotes the minimum mean square prediction error and is defined as

$$P_e \,\square\, \mathrm{E}\left\{\left|e_{FLP}(i)\right|^2\right\}. \tag{2.6}$$

This quantity is also viewed as the ensemble averaged forward prediction error power. If the input samples $u[i]$ have zero mean, then the corresponding forward prediction error would also have zero mean, hence the power under these conditions would be equal to the prediction error variance. As the linear filters are applied in communication systems, they will invariably experience these conditions, hence it is referred to as the prediction error variance σ_e^2.

Letting $\mathbf{p}_0$ denote the optimum tap weight vector which is defined as

$$\mathbf{p}_0 = [p_{01}, p_{02}, \cdots p_{0L}], \tag{2.7}$$

the optimum weights may be solved using the statistical Wiener approach. However, this approach requires two statistics: the cross correlation matrix of the input samples $\mathbf{u}_L$ and the cross correlation vector between the inputs and the desired response $u[i]$. To calculate the prediction error variance, the variance of $u[i]$ is additionally required. Under the assumption that $u[i]$ has zero mean the variance is equal to $r[0]$.

The correlation matrix may be determined by

$$\mathbf{R}_L = \mathrm{E}\left\{\mathbf{u}_L(i-1)\mathbf{u}_L^H(i-1)\right\} = \begin{bmatrix} r[0] & r[1] & \cdots & r[L-1] \\ r[-1] & r[0] & \cdots & r[L-2] \\ \vdots & \vdots & \ddots & \vdots \\ r[-L+1] & r[-L+2] & \cdots & r[0] \end{bmatrix}, \tag{2.8}$$

where $r[k]$ is the autocorrelation of k^{th} lag, which is would be time invariant for a stationary process.

The cross correlation vector is calculated in a similar fashion,

$$\mathbf{r} = \mathrm{E}\left\{\mathbf{u}_L u^*[i]\right\} = \begin{bmatrix} \mathrm{E}\left\{u[i-1]u^*[i]\right\} \\ \mathrm{E}\left\{u[i-2]u^*[i]\right\} \\ \vdots \\ \mathrm{E}\left\{u[i-L]u^*[i]\right\} \end{bmatrix} = \begin{bmatrix} r[-1] \\ \mathrm{r}[-2] \\ \vdots \\ r[-L] \end{bmatrix}. \tag{2.9}$$

Therefore after defining these quantities the normal equation is now modified to

$$\mathbf{R}_L\mathbf{p}_0 = \mathbf{r}, \tag{2.10}$$

which when compared to Yule-Walker equations for the solution of an AR model, it can be seen that they are mathematically of the same form. Due to this mathematical equivalence the solution to the Yule-Walker equations can be applied to solve for the coefficients of a forward linear predictor. The forward prediction error variance may be expressed as

$$P_e = r[0] - \mathbf{r}^H\mathbf{p}_0 = \sigma_u^2 - \mathbf{r}^H\mathbf{p}_0. \tag{2.11}$$

A forward prediction error filter estimates the error in the estimate rather than the estimate itself. The forward error predictor is a new transversal filter with an additional tap weight, however, the order of the filter is still regarded as that of the linear predictor within as seen in Fig. 2.3, since the first tap weight is one.

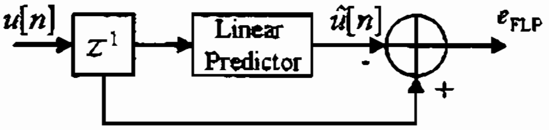

Fig. 2.3 Forward error predictor showing relationship with linear predictor

The forward error predictor may be written in the form of an augmented normal equation. Using (2.10) and (2.11) the augmented form is

$$\begin{bmatrix} r[0] & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R}_L \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{p}_0 \end{bmatrix} = \begin{bmatrix} P_e \\ \mathbf{0} \end{bmatrix}. \tag{2.12}$$

These linear predictor forms are important in understanding the DFDD and prediction based DFDD structures, which will be covered in detail in the next chapter.

2.3.4 The Recursive Least Squares Algorithm

The recursive least squares (RLS) algorithm is one of the most commonly used adaptive recursive algorithms. Its use dates as far back as Gauss (1809). However, its derivation and use has been done independently by many people.

As mentioned earlier the RLS algorithm may be regarded as the deterministic counterpart of the Kalman filter. The RLS algorithm yields an optimal filter in the mean square sense, for each set of input data. The advantage is that it also takes into account the history of the input samples. The rate of convergence of the RLS algorithm is typically an order faster than the simple LMS algorithm. The improvement in performance is of course traded for an increase in computational complexity.

This summary for the RLS algorithm is based on [41], which includes the terms and partial derivation. Generally for recursive implementations the adaptation starts from some known initial conditions, and the filter is used to minimize the performance index $\xi(n)$ (the variance of the estimation error), where n is the variable length of observable data. In the exponentially weighted RLS algorithm an exponential weighting factor (or forgetting factor) is introduced into the definition of the performance index. The performance index is now

$$\xi(n) = \sum_{i=1}^{n} \beta(n,i)\left|e(i)\right|^2 = \sum_{i=1}^{n} \beta(n,i)\left|d_r(i) - y(i)\right|^2, \tag{2.13}$$

where $\beta(n,i)$ is the weighting factor and $e(i)$ is the difference between the desired response $d_r(i)$ and the estimated output $y(i)$ at time index i, over the observation interval n. It is the same structure as that of a forward linear error predictor. However, instead of computing the error between known inputs to estimated outputs, a known desired response is compared to the linear estimate of the desired response. The forgetting factor $\beta(n,i)$ is an exponential forgetting factor and is defined as

$$\beta(n,i) = \Lambda^{n-i},\ i = 1,2,\dots,n\ , \tag{2.14}$$

where $0 < \Lambda \leq 1$, and is generally close to 1 due to the exponential relation. The role of the forgetting factor is important when the environment is non-stationary and tracking is required. The forgetting factor limits the effect of earlier observed data. In general for a stationary environment the forgetting factor is 1, while for a non-stationary environment the forgetting factor is less than 1. The inverse of $1-\Lambda$ is regarded as the memory of the algorithm.

Defining the exponential forgetting factor the performance index is written as

$$\xi(n) = \sum_{i=1}^{n} \Lambda^{n-i} |e(i)|^2 = \sum_{i=1}^{n} \Lambda^{n-i} |d_r(i) - y(i)|^2\ . \tag{2.15}$$

The optimal value of the tap weight vector $\hat{\mathbf{w}}(n)$, is obtained by solving the normal equation

$$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \boldsymbol{\theta}(n), \tag{2.16}$$

where $\mathbf{R}(n)$ is the autocorrelation matrix (ACM) defined as

$$\mathbf{R}(n) = \sum_{i=1}^{n} \Lambda^{n-i} \mathbf{u}(i)\mathbf{u}^H(i), \tag{2.17}$$

and $\boldsymbol{\theta}(n)$ is the cross correlation vector between the tap inputs and the desired response defined as

$$\boldsymbol{\theta}(n) = \sum_{i=1}^{n} \Lambda^{n-i} \mathbf{u}(i) d_r^{*}(i)\ . \tag{2.18}$$

Isolating the term corresponding to $i = n$, equations (2.17) and (2.18) may be re-written in a time recursive fashion as

$$\mathbf{R}(n) = \Lambda\left(\sum_{i=1}^{n-1} \Lambda^{n-1-i}\mathbf{u}(i)\mathbf{u}^{H}(i)\right) + \mathbf{u}(n)\mathbf{u}^{H}(n) = \Lambda\mathbf{R}(n-1) + \mathbf{u}(n)\mathbf{u}^{H}(n) \qquad (2.19)$$

and

$$\boldsymbol{\theta}(n) = \Lambda\boldsymbol{\theta}(n-1) + \mathbf{u}(n)d_r^{*}(n) \qquad (2.20)$$

respectively.

The derivation of the RLS algorithm is based on a theorem in matrix algebra known as the matrix inversion lemma (described in [41]). Applying the lemma to the recursive expression (2.19), the Kalman gain factor $k(n)$, and the tap weight time update $\alpha(n)$ may be derived as shown in [41]. These parameters and their role in the RLS algorithm are shown below.

2.3.4.1 Summary of exponentially weighted RLS Algorithm

- Initialize the algorithm
  - ○ $\mathbf{J}(0) = \Omega^{-1}\mathbf{I}$, Ω is a small positive constant
  - ○ $\hat{\mathbf{w}}(0) = \mathbf{0}$
- For each time instant after initialization compute
  - ○ $\mathbf{k}(n) = \dfrac{\Lambda^{-1}\mathbf{J}(n-1)\mathbf{u}(n)}{1 + \Lambda^{-1}\mathbf{u}^{H}(n)\mathbf{J}(n-1)\mathbf{u}(n)}$
  - ○ $\alpha(n) = d_r(n) - \hat{\mathbf{w}}^{H}(n-1)\mathbf{u}(n)$
  - ○ $\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^{*}(n)$
  - ○ $\mathbf{J}(n) = \Lambda^{-1}\mathbf{J}(n-1) - \Lambda^{-1}\mathbf{k}(n)\mathbf{u}^{H}(n)\mathbf{J}(n-1)$

These four equations are the recursive relations of the RLS algorithm. The initialization of the RLS algorithm will in general be as stated in the summary, unless modified to a particular application. However, it is noted that $\mathbf{J}(0)$ is not explicitly defined. There are two approaches to initialize $\mathbf{J}(0)$ the exact approach or the approximation approach.

The exact approach would use a small segment of the input data because of the relation $\mathbf{J}(n) = \mathbf{R}^{-1}(n)$. The correlation of a few input samples may be used. However, due to the environment it may not be possible to determine the number of samples needed to ensure the invertibility of the matrix. Hence the approximation method is generally applied. In the approximation method $\mathbf{J}(0) = \Omega^{-1}\mathbf{I}$, hence an appropriate Ω needs to be chosen. Intuitively the amount of information gathered in the beginning of the algorithm is vague, hence the inverse correlation matrix is expected to be large; as a rule of thumb, $\Omega^{-1} > 100\sigma_u^2$, where σ_u^2 is the variance of the input samples. However, for large data lengths, the choice of Ω is of no consequence.

2.3.4.2 Convergence Properties of the RLS algorithm

The RLS algorithm converges to a solution in the mean square sense approximately after $2L$ iterations, where L is the order of the transversal filter. Hence it is almost an order of magnitude faster than the LMS algorithm

The convergence rate of the RLS algorithm is relatively insensitive to variations in the eigenvalue spread of the correlation matrix of the input samples $\mathbf{u}(n)$, unlike the LMS algorithm.

Assuming infinite memory ($\Lambda = 1$), as the number of iterations approach infinity the RLS algorithm in theory produces zero excess mean square error, which implies zero misadjustment, therefore it approaches the Wiener solution.

It is noted that the improvement of the convergence rate is inversely proportional to the measurement error, so if the SNR is high the measurement error is small relative to the desired response and hence rapid convergence.

2.3.5 The QR Factorization

The QR factorization is a linear process applied to any matrix. It is used to solve linear systems, and more importantly linear least squares problems. It is used in the new adaptive algorithm, which will be introduced in chapter 4.

The QR factorization, also called the QR decomposition, may be applied to any $m \times n$ matrix $\mathbf{A}$, which may be decomposed as

$$\mathbf{A} = \mathbf{QR} \tag{2.21}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are an $m \times m$ unitary matrix and an $m \times n$ upper triangular respectively. One of the most important applications of the QR factorization is to solve full rank least squares problems, when $\mathbf{Ax} \approx \mathbf{b}$, where $m > n$ and the columns of $\mathbf{A}$ are linearly independent. The solution $\hat{\mathbf{x}}$ that minimizes $\|\mathbf{Ax} - \mathbf{b}\|$ is $\hat{\mathbf{x}} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}$. However, the condition number[†] of $(\mathbf{A}^H\mathbf{A})$ is the square of the condition number of $\mathbf{A}$, hence it would not be advisable to compute $\hat{\mathbf{x}}$ directly. In these cases the QR decomposition is useful, since it mitigates this ill conditioning, by transforming $\mathbf{A}$ into $\mathbf{QR}$.

In some texts the unitary matrix $\mathbf{Q}$ may actually be referring to $\mathbf{Q}^H$, e.g., in [41]. However, depending on the context in which it is used, it may be appropriate to use one or the other. For now the convention used in [42] will be used which is the same as that in (2.21).

Where $m > n$ so the matrix $\mathbf{A}$ is full rank, it may written as

$$\mathbf{A} = \mathbf{QR} = \mathbf{Q}\begin{bmatrix} \mathbf{R}_q \\ \mathbf{0} \end{bmatrix}, \tag{2.22}$$

where $\mathbf{R}_q$ is the $n \times n$ upper triangular matrix, and $\mathbf{0}$ is an $(m-n) \times n$ null matrix.

Then taking the squared Euclidean norm of the least squares problem

$$\|\mathbf{Ax} - \mathbf{b}\|^2 = \|\mathbf{QRx} - \mathbf{b}\|^2 = \|\mathbf{QRx} - \mathbf{QQ}^H\mathbf{b}\|^2 = \|\mathbf{Q}(\mathbf{Rx} - \mathbf{Q}^H\mathbf{b})\|^2, \tag{2.23}$$

which follows because of the unitary matrix definition $\mathbf{QQ}^H = \mathbf{I}$ (where $\mathbf{I}$ is the identity matrix).

[†] The condition number is a measure of how ill conditioned a matrix is. The formal definition is given in chapter 4.

If we let

$$\mathbf{Q}^H\mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix}, \tag{2.24}$$

where $\mathbf{c}$ and $\mathbf{e}$ is an $n \times 1$ and an $(m-n)\times 1$ vector respectively. Utilizing the pre-multiplication property of a unitary matrix: if $\mathbf{y} = \mathbf{Qx}$, where $\mathbf{Q}$ is a unitary matrix then $\|\mathbf{y}\| = \|\mathbf{x}\|$, then (2.23) may be simplified into:

$$\|\mathbf{Ax} - \mathbf{b}\|^2 = \left\| \begin{bmatrix} \mathbf{R}_q \\ \mathbf{0} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix} \right\|^2, \tag{2.25}$$

where (2.24) was substituted, and $\mathbf{Q}$ was removed. This may then be rewritten as

$$\|\mathbf{Ax} - \mathbf{b}\|^2 = \|\mathbf{R}_q\mathbf{x} - \mathbf{c}\|^2 + \|\mathbf{e}\|^2 \tag{2.26}$$

The solution that minimizes (2.26) is the value $\hat{\mathbf{x}}$ which satisfies

$$\mathbf{R}_q\hat{\mathbf{x}} = \mathbf{c}. \tag{2.27}$$

This can be readily computed since $\mathbf{R}_q$ is upper triangular. This least square solution is summarized from [42]. The QR decomposition will not be able to solve this problem directly if $\mathbf{A}$ does not have full rank, i.e. when $m < n$, in these cases alternate methods would have to be used, however, it is recommended in [42] that the single value decomposition be used.

2.3.5.1 Computation of QR Decomposition

There are numerous ways of calculating the QR decomposition, which vary in terms of accuracy and complexity. Some methods also suit applications to certain algorithms, e.g. RLS with the QR decomposition. Generally the most commonly used methods for computing the QR decomposition (as shown in [42]), are:

- The Gram-Schmidt algorithm
- The modified Gram-Schmidt algorithm
- The Householder transformation
- The sequence of Givens rotations.

The Gram Schmidt procedures are the simplest methods to implement. They are also the most inaccurate, therefore a discussion of these procedures is not covered here (for a detailed discussion see section 2.15 in [42]). The Householder transformation attempts to nullify entire columns to achieve the upper triangular matrix by using a suitable reflection operation. The Givens rotations apply a sequence of plane rotations to create the upper triangular matrix one element at a time.

2.3.5.2 The Givens Rotations

The Givens rotations as shown in [41], multiply an $n \times n$ unitary plane rotation matrix $\mathbf{G}$ with an $n \times m$ input matrix $\mathbf{A}$ ($n > m$), to zero an element in an attempt to achieve the $n \times m$ upper triangular matrix $\mathbf{R}$. The plane rotation matrix is defined as follows

$$g_{i,k} = \begin{cases} \cos\phi, \ i,k = m \\ \sin\phi e^{j\theta}, \ i = m, k = n \\ -\sin\phi e^{-j\theta}, \ i = n, k = m \\ \cos\phi, \ i,k = n \\ 1, \ i,k = 1,2,\ldots,n-1, \text{ excluding } m \\ 0, \text{ otherwise.} \end{cases} \tag{2.28}$$

The product in matrix form is

$$\mathbf{GA} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \cos\phi & \cdots & \sin\phi e^{j\theta} \\ \vdots & \vdots & & \vdots \\ 0 & -\sin\phi e^{-j\theta} & \cdots & \cos\phi \end{bmatrix} \begin{bmatrix} & \vdots & \\ \cdots & a_{mm} & \cdots \\ & \vdots & \\ \cdots & a_{nm} & \cdots \end{bmatrix} \tag{2.29}$$

where a_{mm} and a_{nm} represent elements in position (m,m) and (n,m) of the matrix, respectively.

From (2.29), it can be seen that the m$^{\text{th}}$ element in the last row of the product is nulled if the angles θ and ϕ satisfy the condition

$$-\sin\phi e^{-j\theta} a_{mm} + \cos\phi a_{nm} = 0 \tag{2.30}$$

Noting that $\cos^2\phi + \sin^2\phi = 1$, and the real and imaginary parts of the left hand side of (2.30), have to equate to zero respectively, it is determined that

$$\cos\phi = \frac{|a_{mm}|}{\sqrt{|a_{nm}|^2 + |a_{mm}|^2}} \tag{2.31}$$

and

$$\sin\phi e^{j\theta} = \left(\frac{a_{nm}^*}{a_{mm}^*}\right)\cos\phi. \tag{2.32}$$

Using these relations, elements are annihilated from the first column to the last column, last row to uppermost row. Each element annihilated will have an associated unitary rotation matrix, so let this rotation matrix be denoted by $\mathbf{G}(i,k)$, where (i,k) represents the element position nulled.

As a typical example, let $\mathbf{A}$ be a 4×2 matrix therefore the Givens sequence will be

$$\begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix} \xrightarrow{\mathbf{G}(4,1)} \begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ 0 & \times \end{bmatrix} \xrightarrow{\mathbf{G}(3,1)} \begin{bmatrix} \times & \times \\ \times & \times \\ 0 & \times \\ 0 & \times \end{bmatrix} \xrightarrow{\mathbf{G}(2,1)} \begin{bmatrix} \times & \times \\ 0 & \times \\ 0 & \times \\ 0 & \times \end{bmatrix} \xrightarrow{\mathbf{G}(4,2)} \begin{bmatrix} \times & \times \\ 0 & \times \\ 0 & \times \\ 0 & 0 \end{bmatrix} \xrightarrow{\mathbf{G}(3,2)} \begin{bmatrix} \times & \times \\ 0 & \times \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Hence in this case the decomposition becomes $\mathbf{G}(4,1)\mathbf{G}(3,1)\mathbf{G}(2,1)\mathbf{G}(4,2)\mathbf{G}(3,2)\mathbf{A} = \mathbf{R}$, therefore from the definition of the decomposition $\mathbf{A} = \mathbf{QR}$, the unitary matrix becomes

$$\mathbf{Q} = (\mathbf{G}(4,1)\mathbf{G}(3,1)\mathbf{G}(2,1)\mathbf{G}(4,2)\mathbf{G}(3,2))^H. \tag{2.33}$$

2.4 Adaptive Techniques for Non-coherent Detection

A number of attempts have been made to employ adaptive filter theory (AFT) to improve the performance of non-coherent detection. They have been successful to varying degrees. Some of the adaptive schemes show promising performance with minimal penalties in terms of information capacity and complexity. A few of these adaptive detection schemes are discussed in this section.

2.4.1 Adaptive Channel Estimation

One of the early attempts to improve the performance of differential detection in fast fading environments was introduced in [43]. Here the channel gain was estimated using multiple delays, in conjunction with an adaptive algorithm, hence the name: adaptive channel estimation or the ACE detector.

This scheme focused on continuous phase modulation (CPM) with minimum shift keying (MSK). ACE offered significant improvement over differential MSK in both AWGN and fast fading channels. An improvement of an order was evident for channels with normalized bandwidths of $F_dT = 0.05$ and $F_dT = 0.1$. The ACE detector utilized the recursive least squares lattice (RLSL) algorithm, which is numerically stable, modular and fast converging. However, the filter order would be relatively high, generally ten or more delays are necessary for the performance gains shown.

2.4.2 Decision Feedback Adaptive Linear Prediction

An alternate adaptive technique which performed better than SDD was introduced in [44]. This algorithm involved decision feedback of the actual channel estimate. The LMS algorithm was the adaptive algorithm used. The LMS algorithm is not ideal, with slow convergence. A training period was used, which was not bandwidth efficient, and the improvement over SDD is only evident if the training period is ignored. In addition to being slow converging, the order of the filter used was high and was not optimal over a range of normalized Doppler frequencies. At a frequency of $F_dT = 0.05$, the required filter order was approximately 50, hence this algorithm is not suited for use in fast fading environments.

2.4.3 DFDD with RLS

This scheme involves decision feedback of symbols, and not the channel estimate. In [22], an analogous system to DFDD over flat Rayleigh fading channels was introduced. Instead of the original DFDD which was derived specifically for Rayleigh fading channels, an alternate system based on linear prediction was used. The filter coefficients were based on the Yule-Walker equations.

In the same publication an adaptive scheme using the RLS algorithm was also introduced. Both the adaptive and linear predication based scheme was found to be equivalent in performance to the DFDD for Rayleigh flat fading environments. It should be noted that this scheme was introduced for Ricean fading, and the decision metric was not derived from the optimal maximum likelihood probability metric, unlike that of DFDD for flat fading channels. A Ricean channel may be regarded as an AWGN or Rayleigh channel under special circumstances. This scheme will also be discussed in greater detail in chapter 3, since it is later applied to the original DFDM scheme as part of this dissertation's contribution.

2.4.4 Adaptive Detection and Decoding via Mixture Kalman Filtering

Recently an alternate scheme was introduced by Chen and Wang in [45], which can be utilized for both coded and uncoded cases. The receiver uses sequential Monte Carlo methodology for the adaptive Bayesian receiver. The scheme is self adaptive, with no training necessary, while the receiver structure is well suited for VLSI.

The performance of this receiver approached the theoretical bounds, and significantly improved on SDD in flat fading environments. There is additional discussion in [46], regarding the mixture Kalman filtering employed as the adaptive algorithm.

Chapter 3
Fundamental Decision Feedback Schemes

Standard differential detection (SDD) of DM-PSK signals does not require the current channel state information, nor does it require the channel statistical characteristics (e.g. autocorrelation, noise and fading variance). However, as mentioned in chapter 2, in frequency flat channels with high Doppler frequencies, an error floor is incurred that would not allow reliable transmission of information. Therefore in an attempt to reduce this error floor, MSD with decision feedback [21], was a technique introduced, which virtually eliminated the error floor while remaining computationally simple. This scheme is known as DFDD.

For optimal performance of DFDD it is necessary to know the channel characteristics: the noise variance and the correlation of the fading channel. If these statistics are unknown at the receiver, sub-optimal performance would be achieved, which would still be superior to SDD in most cases. Adaptive filtering mitigates the need for any channel knowledge, even the second order channel statistics, while still achieving performance comparable to the optimal fixed solution. One such structure employing the RLS algorithm was introduced in [22], which is referred to as adaptive DFDD. This adaptive DFDD has the same feedback structure as the DFDD detector of [21], and is based on the linear prediction based DFDD introduced in [22].

In addition to these DFDD systems, the iterative coded DFDM scheme of [23] is discussed. The DFDM utilizes the DFDD decision metric of [21] as a probability metric. The DFDM is derived and the related performance analysis from [23] discussed.

In section 3.1, the original DFDD system for Rayleigh flat fading channels of [21] is introduced. The alternate DFDD system and its adaptive counterpart of [22], are discussed in section 3.2. The iterative coded DFDM scheme of [23] and the associated convergence and analysis are described in section 3.3. The performance of these uncoded DFDD detectors, and the iterative DFDM are replicated for comparison purposes in section 3.4.

3.1 Decision Feedback Differential Detection (DFDD)

This section summarizes the DFDD scheme for flat fading channels introduced in [21], with relevant derivations and definitions shown.

3.1.1 System model

The block diagram of the DFDD system is shown in Fig. 3.1. The information bits are modulated to the M-PSK constellation set for any constellation size M=2, 4, 8 and 16. The constellation set $\Upsilon \square \{\exp(j(2\pi v/M)) \mid v \in \{0,1,\ldots,M-1\}\}$, would represent both the modulated symbols $a[k]$ as well as the differentially encoded symbols $d[k]$. The differentially encoded symbols are given by

$$d[k] = a[k]d[k-1],\ k \in \square, \tag{3.1}$$

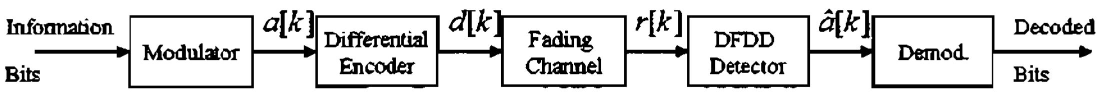

Fig. 3.1 The block diagram of the DFDD detector.

As it is only necessary to utilize the baseband signals the equivalent carrier signals are ignored. The fading channel is modelled as time varying Rayleigh frequency flat fading. The mobility of the channel is characterized by the maximum Doppler frequency F_d. The Doppler shifts may be normalized with the symbol period T, to yield the normalized fading bandwidth F_dT. The symbols at the receiver are scaled by the fading gain and perturbed by noise which can be written as

$$r[k] = d[k]f[k] + n[k] \tag{3.2}$$

where the fading process $f[\square]$, and the noise samples $n[\square]$ are correlated and mutually independent zero mean complex Gaussian random processes, respectively. It is also assumed that $f[\square]$ and $n[\square]$ are mutually uncorrelated. Due to normalization, $f[\square]$ and

$n[\square]$ have variance $\sigma_f^2 = \mathrm{E}\{|f[k]|^2\} = 1$ and $\sigma_n^2 = \mathrm{E}\{|n[k]|^2\} = N_0 / E_S$, where N_0 and E_S denote the one sided power spectral density of the AWGN process and the mean symbol energy, respectively. The received sequence is deciphered using the DFDD detector and demodulated (denoted as Demod. in Fig. 3.1) into the equivalent bits to represent the original information sequence.

3.1.2 Derivation of DFDD Metric

The DFDD metric is based on the optimum MSD metric. The optimum MSD metric is derived using an observation window over N symbol periods, i.e., NT, where T is the period of a symbol and a length constraint of $N > 1$ is used. Considering that an observation interval is used, vectors are defined accordingly. Based on (3.2), the vector equation is

$$\mathbf{r}_k = \mathbf{d}_k \mathbf{f}_k + \mathbf{n}_k \tag{3.3}$$

where,

$$\mathbf{r}_k \,\square\, \left[r[k], r[k-1], \ldots, r[k-N+1]\right]^T, \tag{3.4}$$

$$\mathbf{d}_k \,\square\, \mathrm{diag}\left\{d[k], d[k-1], \ldots, d[k-N+1]\right\}, \tag{3.5}$$

$$\mathbf{f}_k \,\square\, \left[f[k], f[k-1], \ldots, f[k-N+1]\right]^T, \tag{3.6}$$

$$\mathbf{n}_k \,\square\, \left[n[k], n[k-1], \ldots, n[k-N+1]\right]^T, \tag{3.7}$$

and $\mathrm{diag}\{x[1], x[2], \ldots, x[L]\}$ denotes an $L \times L$ diagonal matrix with the diagonal elements comprised of the elements $\{x[1], x[2], \ldots, x[L]\}$, respectively.

In matrix form (3.3) is as follows

$$\begin{bmatrix} r[k] \\ r[k-1] \\ \vdots \\ r[k-N+1] \end{bmatrix} = \begin{bmatrix} d[k] & 0 & \cdots & 0 \\ 0 & d[k-1] & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d[k-N+1] \end{bmatrix} \begin{bmatrix} f[k] \\ f[k-1] \\ \vdots \\ f[k-N+1] \end{bmatrix} + \begin{bmatrix} n[k] \\ n[k-1] \\ \vdots \\ n[k-N+1] \end{bmatrix} \tag{3.8}$$

The conditional probability density function (pdf) $p(\mathbf{r}_k \,|\, \mathbf{a}_k)$ is

$$p(\mathbf{r}_k \mid \mathbf{a}_k) = \frac{1}{\pi^N |\mathbf{R}_a|_{\text{det}}} \exp\left(-\mathbf{r}_k^H \mathbf{R}_a^{-1} \mathbf{r}_k\right), \tag{3.9}$$

where $\mathbf{R}_a$ is the $N \times N$ conditional autocorrelation matrix of $\mathbf{r}_k$ defined as

$$\mathbf{R}_a \square \mathrm{E}\left\{\mathbf{r}_k \mathbf{r}_k^H \mid \mathbf{a}_k\right\} \tag{3.10}$$

Defining the autocorrelation of the fading process as

$$\mathbf{R}_f \square \mathrm{E}\left\{\mathbf{f}_k \mathbf{f}_k^H\right\} \tag{3.11}$$

the autocorrelation $\mathbf{R}_a$ is re-written as

$$\mathbf{R}_a = \mathbf{d}_k \mathbf{R}_f \mathbf{d}_k^H + \sigma_n^2 \mathbf{I} \tag{3.12}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix. The maximum likelihood decision for $\mathbf{a}_k$ corresponding to MSD can be obtained by maximizing (3.9). This is equivalent to maximizing the metric

$$\eta = -\mathbf{r}_k^H \mathbf{R}_a^{-1} \mathbf{r}_k - \ln |\mathbf{R}_a|_{\text{det}}. \tag{3.13}$$

$\mathbf{R}_a$ can be expressed in another form due to the relation $\mathbf{d}_k \mathbf{d}_k^H = \mathbf{I}$, substituting this relation into (3.12),

$$\mathbf{R}_a = \mathbf{d}_k \left(\mathbf{R}_f + \sigma_n^2 \mathbf{I}\right) \mathbf{d}_k^H \square \mathbf{d}_k \mathbf{R} \mathbf{d}_k^H. \tag{3.14}$$

Thus the determinant of $\mathbf{R}_a$ may also be re-written as

$$|\mathbf{R}_a|_{\text{det}} = |\mathbf{d}_k|_{\text{det}} |\mathbf{R}|_{\text{det}} |\mathbf{d}_k^H|_{\text{det}} = |\mathbf{d}_k|_{\text{det}} |\mathbf{R}|_{\text{det}} |\mathbf{d}_k|_{\text{det}}^H = |\mathbf{R}|_{\text{det}}, \tag{3.15}$$

where it can be seen that the determinant is independent of the transmitted symbol sequence. Knowing $\mathbf{d}_k \mathbf{d}_k^H = \mathbf{I}$, this means $\mathbf{d}_k^H = \mathbf{d}_k^{-1}$, therefore the first term in (3.13) of the ML metric can be simplified to

$$\eta' = -\mathbf{r}_k^H \mathbf{d}_k \mathbf{R}^{-1} \mathbf{d}_k^H \mathbf{r}_k. \tag{3.16}$$

Defining the negative inverse correlation matrix $\mathbf{T}$ as

$$\mathbf{T} = -\mathbf{R}^{-1} \square \begin{bmatrix} t_{00} & t_{01} & \cdots & \cdots & t_{0(N-1)} \\ t_{10} & t_{11} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{(N-2)(N-2)} & t_{(N-2)(N-1)} \\ t_{(N-1)0} & \cdots & \cdots & t_{(N-1)(N-2)} & t_{(N-1)(N-1)} \end{bmatrix}, \tag{3.17}$$

and noting that since $\mathbf{R}$ is Hermitian, its inverse would hold the same property, therefore $t_{uv} = t_{vu}^{*}$ for $0 \le v,\ u \le N-1$.

Equation (3.16) can be written as

$$\eta' = \sum_{v=0}^{N-1} \sum_{u=0}^{N-1} t_{vu} d[k-v] d^{*}[k-u] r^{*}[k-v] r[k-u] \tag{3.18}$$

determined through some basic linear algebra. However, this metric can be further simplified as

$$\eta' = \sum_{v=0}^{N-1} t_{vv} \left| d[k-v] \right|^{2} \left| r[k-v] \right|^{2} + 2 \cdot \mathrm{Re}\left\{ \sum_{v=0}^{N-1} \sum_{u=v+1}^{N-1} t_{vu} r[k-u] r^{*}[k-v] \prod_{j=v}^{u-1} a[k-j] \right\} \tag{3.19}$$

where the difference relation of equation (3.1) has been applied. From (3.19) it can be seen that the decision to determine the estimated symbol vector $\hat{\mathbf{a}}_k \square [\hat{a}[k], \hat{a}[k-1], \ldots, \hat{a}[k-N+2]]^{T}$, $\hat{a}[\square] \in \Upsilon$, is determined by the second term, therefore

$$\hat{\mathbf{a}}_k = \arg\max_{\tilde{\mathbf{a}}_k} \left\{ \mathrm{Re}\left\{ \sum_{v=0}^{N-1} \sum_{u=v+1}^{N-1} t_{vu} r[k-u] r^{*}[k-v] \prod_{j=v}^{u-1} \tilde{a}[k-j] \right\} \right\}, \tag{3.20}$$

where the vector of unknown transmitted symbols are replaced by a vector of trial symbols $\tilde{\mathbf{a}}_k \square [\tilde{a}[k], \tilde{a}[k-1], \ldots, \tilde{a}[k-N+2]]^{T}$, $\tilde{a}[\square] \in \Upsilon$, and $\hat{\mathbf{a}}_k = \arg\max_{\tilde{\mathbf{a}}_k}\{\square\}$ denotes the vector $\hat{\mathbf{a}}_k = \tilde{\mathbf{a}}_k$ that maximizes the function in parentheses.

However, this is a block level metric, which can be simplified by using decision feedback symbols (symbols already deciphered) $\hat{a}[k-v]$ instead of the trial symbols in the decision metric (3.20). By doing this, the block level implementation becomes a symbol level implementation, i.e., a decision is made for only one symbol at a time.

Omitting all terms that do not influence the symbol decision the decision metric becomes

$$\hat{a}[k] = \arg\max_{\tilde{a}[k]} \left\{ \mathrm{Re}\left\{ \tilde{a}^*[k] r[k] \left(\sum_{v=1}^{N-1} t_{0v} r[k-v] \prod_{j=1}^{v-1} \hat{a}[k-j] \right)^* \right\} \right\}. \tag{3.21}$$

3.1.3 DFDD Detector

The structure of the DFDD detector is shown in Fig. 3.2, where it can be seen that the DFDD detector is an autoregressive (AR) FIR filter of order $N-1$. The Rayleigh frequency flat fading is modelled using the Jakes model.

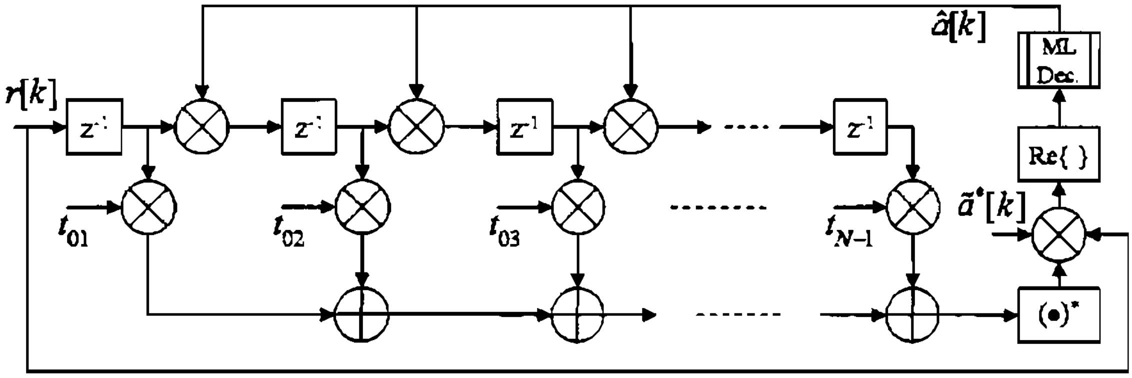

Fig. 3.2 The DFDD detector structure.

For the Jake's model the autocorrelation function (ACF) of the fading process is given by

$$R_f[\tau] = \mathrm{E}\{f^*[k] f[k+\tau]\} = \sigma_f^2 \cdot J_0(2\pi F_d T \tau),\ \forall \tau \tag{3.22}$$

where $J_0(\Box)$ is a zeroth order Bessel function of the first kind. Therefore the autocorrelation matrix (ACM) would have the structure shown below. If, for example, a normalized fading bandwidth of $F_d T = 0.03$ and observation window of $N = 3$ was used the ACM of the fading process would be as follows:

$$\begin{aligned}\mathbf{R}_f &= \begin{bmatrix} J_0(2\pi\cdot 0.03\cdot 0) & J_0(2\pi\cdot 0.03\cdot 1) & J_0(2\pi\cdot 0.03\cdot 2) \\ J_0(2\pi\cdot 0.03\cdot 1) & J_0(2\pi\cdot 0.03\cdot 0) & J_0(2\pi\cdot 0.03\cdot 1) \\ J_0(2\pi\cdot 0.03\cdot 2) & J_0(2\pi\cdot 0.03\cdot 1) & J_0(2\pi\cdot 0.03\cdot 0) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0.9911 & 0.9648 \\ 0.9911 & 1 & 0.9911 \\ 0.9648 & 0.9911 & 1 \end{bmatrix}\end{aligned} \qquad (3.23)$$

Knowing this the negative inverse of the ACM ($\mathbf{R} = \mathbf{R}_f + \sigma_n^2 \mathbf{I}_N$) may be solved and used for the filter coefficients in the DFDD metric.

The ACM is symmetric, this implies that the corresponding inverse has the same property, and hence the first row or the first column may be used interchangeably for the decision, and $t_{0v} = t_v$, for $1 \le v \le N-1$. Therefore the expression becomes

$$\hat{a}[k] = \underset{\tilde{a}[k]}{\arg\max}\left\{ \mathrm{Re}\left\{ \tilde{a}^*[k] r[k] \left(\sum_{v=1}^{N-1} t_v r[k-v] \prod_{j=1}^{v-1} \hat{a}[k-j] \right)^* \right\} \right\} \qquad (3.24)$$

It was noted in [21] that for $N = 2$, the structure and performance of DFDD is equivalent to SDD. The decision is made with only two received symbols, and no feedback. The performance for $N = 2$ also depicts an error floor for large normalized bandwidths in flat fading conditions.

The detector structure as seen in Fig. 3.2, is similar to that of a standard linear predictor, described in chapter 2. In [21], this linear predictor approach proved highly useful for the purpose of analysis, and for understanding the dynamics of the structure. It was seen that the probability of error was dependent on the prediction error variance, the fading variance and the noise variance. When the noise approached zero, the fading variance would dominate the probability, which implies an irreducible error floor.

3.2 Adaptive DFDD

An alternate form of detection using decision feedback and linear prediction was introduced in [22], which was designed to perform in Ricean channels. The Ricean channel incorporates the AWGN channel and the Rayleigh flat fading channel as special cases. This prediction based DFDD may be regarded as the normalized DFDD structure, because it utilizes normalized filter coefficients for the linear predictor.

The normalized filter structure is conducive to adaptive filtering techniques. In [22], the authors introduced an adaptive algorithm utilizing the RLS algorithm. This adaptive DFDD algorithm was introduced as an alternative to the fixed coefficient prediction based DFDD. These two versions of DFDD are derived and discussed in this section.

3.2.1 Prediction based DFDD

The derivation of the DFDD of [21] is derived directly from the ML probability expression, which may be considered to be the optimal DFDD scheme for flat fading channels. The prediction based DFDD is derived differently. In [22], the transmission model is not strictly a Rayleigh fading channel, but it is shown that the system does compare to the DFDD scheme for flat Rayleigh fading channels.

The transmission model in [22] has a frequency offset Δf between the modulator and demodulator included as well as a uniformly distributed phase shift Θ. The received signal sample for differentially encoded symbols become

$$r[k] = e^{j\Theta} e^{j2\pi\Delta f Tk} f[k] d[k] + n[k]. \tag{3.25}$$

The noise process and fading process have normalized energy and variance, where $\sigma_n^2 = \mathrm{E}\{|n[k]|^2\} = N_0 / E_S$ and $\sigma_f^2 = \mathrm{E}\{|f[k]|^2\} = 1$, respectively.

The prediction based DFDD is based on standard differential detection. In SDD, the reference symbol used to make the decision on $a[k]$ is $r[k-1]$, where in general

$$a[k] = r[k] r^*[k-1]. \tag{3.26}$$

It is also noted that (3.25) may be re-written as

$$r[k] = a[k] e^{j\Theta} e^{j2\pi\Delta f Tk} f[k] d[k-1] + n[k], \tag{3.27}$$

hence it is proposed that an estimate of $e^{j\Theta}e^{j2\pi\Delta fTk}f[k]d[k-1]$, is made using not just $r[k-1]$, but the last $N-1$ received samples as well. Hence the estimated symbol may be written as

$$r_e[k-1]=\sum_{v=1}^{N-1}p_{0_v}r[k-v], \tag{3.28}$$

where p_{0_v} is the tap weight coefficients of the linear predictor with $1\le v\le N-1$. Since the noise process is an uncorrelated Gaussian random process, the estimator is equivalent to a linear estimate of $(e^{j\Theta}e^{j2\pi\Delta fTk}f[k]d[k-1]+n[k])/a[k]=r[k]/a[k]$. The mean square error variance σ_{mse}^2, of the estimation is

$$\sigma_{mse}^2=\mathrm{E}\left\{\left|\frac{r[k]}{a[k]}-r_e[k-1]\right|^2\right\}=\mathrm{E}\left\{\left|r[k]-a[k]r_e[k-1]\right|^2\right\}, \tag{3.29}$$

which, in terms of the transmission model may be written as

$$\sigma_{mse}^2=\mathrm{E}\left\{\left|\begin{matrix}e^{j\Theta}e^{j2\pi\Delta fTk}f[k]d[k-1]+n[k]-\\ a[k]\sum_{v=1}^{N-1}p_{0_v}\left(e^{j\Theta}e^{j2\pi\Delta fT(k-v)}f[k-v]d[k-v]+n[k-v]\right)\end{matrix}\right|^2\right\}. \tag{3.30}$$

Using the substitution and the differential encoding relation,

$$p_{0_v}\cong\frac{d[k-1]}{d[k-v]}p_v=\prod_{j=1}^{v-1}a[k-j]p_v, \tag{3.31}$$

this is substituted into (3.30), simplifying it to

$$\sigma_{mse}^2=\mathrm{E}\left\{\left|c[k]-\sum_{v=1}^{N-1}p_vc[k-v]\right|^2\right\}, \tag{3.32}$$

where

$$c[k]\,\Box\,e^{j2\pi\Delta fTk}f[k]+n'[k] \tag{3.33}$$

and

$$n'[k]\,\Box\,\frac{e^{-j\Theta}n[k]}{d[k]}. \tag{3.34}$$

It is noted that $n'[\square]$ is uncorrelated white Gaussian noise with the same variance σ_n^2 as the noise process $n[\square]$.

As shown in (3.28), p_v may be interpreted as the tap weights of an $N-1$ order linear predictor, where the coefficients may be solved using the Yule-Walker equations

$$\mathbf{R}_c\mathbf{p} = \mathbf{r}_c \tag{3.35}$$

where $\mathbf{R}_c$ is the $(N-1)\times(N-1)$ autocorrelation matrix of the process $c[\square]$, $\mathbf{p}$ is the tap weight coefficients, and $\mathbf{r}_c$ is the correlation vector. These quantities are defined as follows

$$R_c[\tau] \square \mathrm{E}\left\{c[k+\tau]c^*[k]\right\} = e^{j2\pi\Delta fT\tau}R_f[\tau] + \sigma_n^2\delta[\tau], \tag{3.36}$$

$$\mathbf{p} \square \left[p_1, p_2, \ldots p_{N-1}\right]^T, \tag{3.37}$$

$$\mathbf{r}_c \square \left[R_c[-1], R_c[-2], \ldots, R_c[-N+1]\right]^T \tag{3.38}$$

and $\mathbf{R}_c$ is the ACM whose structure is based on the ACF as shown in (2.8) and $\delta[\square]$ is the unit pulse sequence (i.e. $\delta[0]=1$, $\delta[\tau]=0$, for $\tau \neq 0$).

Thus far it was assumed that the symbols $a[k-v]$ were perfectly known at the receiver. However, in a practical system this is not possible. Hence decision feedback symbols $\hat{a}[k-v]$ are used for $1 \leq v \leq N-2$. The resulting reference symbol due to decision feedback becomes

$$r_e[k-1] = \sum_{v=1}^{N-1} p_v r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j], \tag{3.39}$$

and the resulting decision metric based on SDD becomes

$$\hat{a}[k] = \underset{\tilde{a}[k]}{\arg\max}\left\{\mathrm{Re}\left\{\tilde{a}[k]r^*[k]r_e[k-1]\right\}\right\}. \tag{3.40}$$

Since Rayleigh frequency flat fading is the required channel, the channel parameters Δf and Θ may be assumed 0. The fading process is modelled using the Jakes model, hence the ACF and ACM of the process $c[\square]$, would be equivalent to the autocorrelation matrix described in standard DFDD.

The process described here is the prediction based DFDD. In this case the channel statistics are known *a priori* and the Yule-Walker equations may be solved directly. However, if they were not then the filter coefficients may be determined adaptively. The adaptive scheme is discussed in the next section.

3.2.2 Adaptive DFDD with RLS algorithm

In many cases the channel statistics in mobile communications are non-stationary hence channel statistics would have to be measured at regular intervals in order to determine them *a priori*. Once the channel statistics are available, prediction based DFDD or the DFDD may be employed, because these algorithms require the channel statistics for optimal performance.

In [22] an adaptive scheme was proposed to compute the linear filter coefficients recursively, using the RLS algorithm. The RLS algorithm is a common, fast converging, and recursive algorithm which is superior to the LMS algorithm in almost every performance criteria. It is however computationally more complex and intermittently suffers from numerical instability.

The associated performance index related to predictive DFDD is

$$\xi[k] = \sum_{j=1}^{k} \Lambda^{k-j} \left| r[j] - \sum_{v=1}^{N-1} p_v[k]\hat{r}[j,v] \right|, \tag{3.41}$$

where $p_v[k]$ are the predictor coefficients at time k and $\hat{r}[k,v]$ is defined as

$$\hat{r}[k,v] \Box\ r[k-v]\prod_{\mu=0}^{v-1} \hat{a}[k-\mu], \quad 1 \le v \le N-1, \tag{3.42}$$

which is taken from (3.39) which is part of the linear estimate of $r_e[k-1]$. The desired response would hence be $r[k-1]$, or depending on the time reference $r[k]$, because the relative time index is merely shifted by one. For the RLS algorithm the desired response is the previous symbol in terms of the decision metric for prediction based DFDD.

Provided all random processes are ergodic (the time averages may be used as the ensemble averages, this implies weakly stationary), for a forgetting factor $\Lambda = 1$ and k approaching ∞, $p_v[k] \rightarrow p_v$, the optimal Wiener solution from the Yule-Walker equations for prediction based DFDD. The equations of the RLS algorithm, from [22], for adaptive DFDD referred to as RLS-DFDD are:

$$\mathbf{k}_k = \frac{\Lambda^{-1}\mathbf{J}_{k-1}\hat{\mathbf{r}}_k}{1+\Lambda^{-1}\hat{\mathbf{r}}_k^H\mathbf{J}_{k-1}\hat{\mathbf{r}}_k} \tag{3.43}$$

$$\alpha[k] = r[k] - \mathbf{p}_{k-1}^T\hat{\mathbf{r}}_k \tag{3.44}$$

$$\mathbf{J}_k = \Lambda^{-1}\mathbf{J}_{k-1} - \Lambda^{-1}\mathbf{k}_k\hat{\mathbf{r}}_k^H\mathbf{J}_{k-1} \tag{3.45}$$

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{k}_k^*\alpha[k] \tag{3.46}$$

where $\hat{\mathbf{r}}_k$ and $\mathbf{p}_k$ are defined as

$$\hat{\mathbf{r}}_k \,\square\, \left[\hat{r}[k,1], \hat{r}[k,2], \ldots, \hat{r}[k,N-1]\right]^T \tag{3.47}$$

and

$$\mathbf{p}_k \,\square\, \left[p_1[k], p_2[k], \ldots, p_{N-1}[k]\right]^T \tag{3.48}$$

respectively. The approximation initialization approach was used hence $\mathbf{J}(0) = \Omega^{-1}\mathbf{I}$, and the initial filter coefficients are

$$\mathbf{p}_0 = [1, 0, 0, \ldots, 0]^T. \tag{3.49}$$

The initial filter coefficients are non-standard to allow the algorithm to start the adaptation process from the conventional differential detection case ($N = 2$). Based on this implementation the adaptation may start blind, with no training symbols, and with no *a priori* channel statistics. As mentioned in the previous section, if the mobile channel is known to be statistically stationary, then $\Lambda = 1$ may be used, however, this is typically not the case so $\Lambda < 1$ should be chosen.

It was noted in [22] that the filter coefficients approached the optimal values, and after 10 iterations the performance for $N = 3, 5$ at typical SNRs, the mean square error of the adaptive receiver was lower than that of $N=2$. Since the BER is dependent on the mean square error, it was expected that the adaptive receiver will outperform SDD.

3.2.3 Equivalence of DFDD and Prediction based DFDD

It should be noted that for Rayleigh fading channels, there is a mathematical equivalence between the original DFDD metric of [21], and that of [22], which was not shown in [22] or any other literature available. Mathematical equivalence; meaning the decisions of both metrics would yield the same result, for the same input samples. The channel statistics are known at the receiver in both cases.

In [21], it was shown that the DFDD detector may be treated as a linear prediction error filter. As shown a linear error predictor has an augmented normal equation structure as in (2.12), so for DFDD it is,

$$\mathbf{R}\begin{bmatrix} 1 \\ -\mathbf{p}^* \end{bmatrix} = \begin{bmatrix} \sigma_e^2 \\ \mathbf{0} \end{bmatrix}, \tag{3.50}$$

where $\mathbf{p}$ is an $N-1\times 1$ vector of filter coefficients defined as

$$\mathbf{p} \square \left[p_1, p_2, \ldots, p_{N-1}\right]^T, \tag{3.51}$$

and σ_e^2 is the prediction error variance, while $\mathbf{R}$ is an $N\times N$ matrix defined as for DFDD in (3.23). The optimum coefficients are related to the inverse of the correlation matrix $\mathbf{R}$. From (3.50) it can be seen that

$$\mathbf{t}_0 = -\frac{1}{\sigma_e^2}\begin{bmatrix} 1 \\ -\mathbf{p}^* \end{bmatrix}, \tag{3.52}$$

where $\mathbf{t}_0$, is defined as the first column of negative inverse correlation matrix $\mathbf{R}$, defined in (3.17). The optimum coefficients in terms of $\mathbf{p}$ are

$$t_0 = -\frac{1}{\sigma_e^2}, \tag{3.53}$$

$$t_v = \frac{p_v}{\sigma_e^2}, \quad 1 \le v \le N-1. \tag{3.54}$$

Hence it may be seen that prediction based DFDD is the normalized linear prediction filter for standard DFDD for a Rayleigh channel, since the autocorrelation function was based on the Jakes model. For convenience the DFDD metric for Rayleigh flat fading channel from [21] is

$$\hat{a}[k] = \arg\max_{\hat{a}[k]} \left\{ \mathrm{Re}\left\{ \bar{a}^*[k] r[k] \left(\sum_{v=1}^{N-1} t_v r[k-v] \prod_{j=1}^{v-1} a[k-j] \right)^* \right\} \right\}, \tag{3.55}$$

while the decision metric for prediction based DFDD is

$$\hat{a}[k] = \arg\max_{\tilde{a}[k]} \left\{ \mathrm{Re}\left\{ \tilde{a}[k] r^*[k] \sum_{v=1}^{N-1} p_v r[k-v] \prod_{j=1}^{v-1} a[k-j] \right\} \right\}. \tag{3.56}$$

Taking the complex conjugate of the inner expression of (3.55) would not alter the decision since only the real part of the expression is required and substituting (3.54) the standard DFDD metric for Rayleigh frequency flat fading becomes

$$\hat{a}[k] = \arg\max_{\tilde{a}[k]} \left\{ \frac{1}{\sigma_e^2} \square \mathrm{Re}\left\{ \tilde{a}[k] r^*[k] \left(\sum_{v=1}^{N-1} p_v r[k-v] \prod_{j=1}^{v-1} a[k-j] \right) \right\} \right\}, \tag{3.57}$$

therefore it can be seen that prediction based DFDD is the normalized DFDD metric and they would both make the same decision.

3.3 Iterative Decision Feedback Differential Demodulation

Thus far the DFDD schemes discussed above did not have channel coding applied. Numerous MSD systems with channel coding were introduced in chapter 2. Among these was iterative decision feedback differential demodulation (DFDM). This relatively simple receiver, utilized the DFDD metric, and the concept of bit interleaved coded modulation (BICM). This iterative receiver improved the performance over the conventional non-coherent BICM in Rayleigh flat fading channels.

This section summarizes the DFDM system from [23]. The derivation of the metrics, as well as the convergence and cut-off rate analysis are covered.

3.3.1 System Model

The discrete time model of the iterative DFDM is shown in Fig. 3.3. It can be seen that the transmission scheme is exactly the same as BICM, however, the detection scheme is iterative in nature.

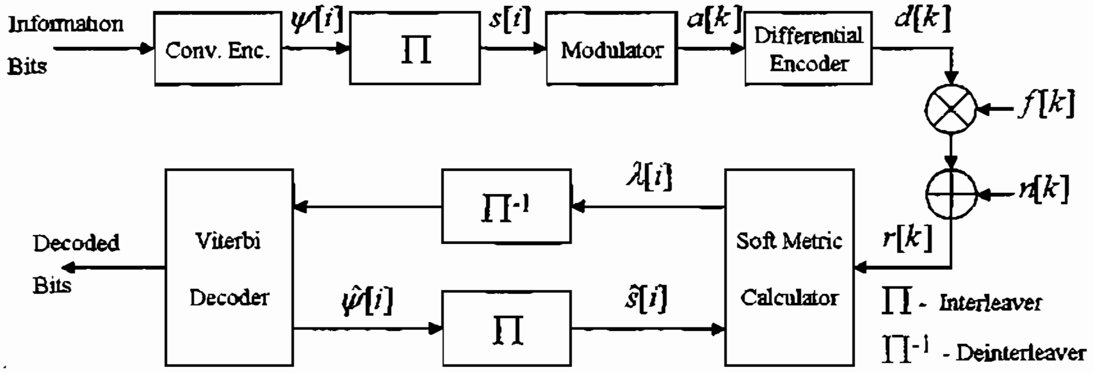

Fig. 3.3 Block Diagram of iterative DFDM

All signals are represented by their baseband equivalents, and all quantities are complex valued. The information bits are encoded using a standard off the shelf convolutional code, into coded bits denoted by $\psi[i]$. These coded bits are randomly permuted when passed through a bit interleaver yielding a shuffled coded bit sequence $s[i]$. A standard MPSK constellation Υ, defined as in DFDD, is assumed with Gray labelling. The permuted sequence is mapped to the constellation set Υ, via an appropriate mapping function $X\{\square\}$. The modulated symbols $a[k]$ are then differentially encoded to represent the channel symbols $d[k]$. The standard differential relation for PSK is used:

$$d[k] = a[k]d[k-1] \tag{3.58}$$

The random fading process $f[\square]$ is mutually independent and correlated, modelled according to Jakes [47], with variance σ_f^2. The random noise process $n[\square]$ is mutually uncorrelated zero mean complex Gaussian with variance σ_n^2.

Under these condition the received symbol is represented as

$$r[k] = f[k]d[k] + n[k], \tag{3.59}$$

The iterative receiver is discussed in a subsequent section.

3.3.2 DFDM Metric Derivation

From BICM the soft bit metrics, $\lambda[i]$ are computed according to the conditional probability density function (pdf) $p(\mathbf{r}_k \mid \mathbf{a}_k)$. However, the soft metric used in DFDM is based on MSD and hence the soft metric is determined using observation windows.

Assuming all data symbols are a priori equally probable, the probability density function is given by:

$$p(\mathbf{r}_k \mid \mathbf{a}_k) = p(r[k], r[k-1], \ldots, r[k-N+1] \mid a[k], a[k-1], \ldots, a[k-N+2]). \quad (3.60)$$

From this it can be seen that if trial sequences are used for $\mathbf{a}_k$, then the complexity is exponentially related to N. Instead of the set of all possible vectors (for e.g. [48]), which would yield the optimal performance, the computationally less complex case of decision feedback symbols are used, analogous to DFDD.

The soft bit metric $\lambda[i]$ may be determined through the use of hard decision feedback symbols $\hat{a}[k-v]$, $v = 1, 2, \ldots, N-2$, leaving only $a[k]$ as a trial symbol, which is exactly the metric used in DFDD. If M is the size of the constellation, then $l = \log_2 M$ is the number of bits used to represent the modulated symbol. An index $\ell = 0, 1, \ldots, l-1$, is used to denote the position of the bit, where it can be determined that $i = kl + \ell$. Due to the use of observation windows, and only one trial symbol, the time index k, is of no importance. The metric notation will be determined by the type of feedback (*sym* or *bit*) used, the value of the bit ($b \in \{0,1\}$), and the bit position in the trial symbol. Hence i may be replaced by ℓ, and the metric may be written as

$$\lambda_b^{sym}[\ell] = \log \sum_{a[k] \in \Upsilon_b^{\ell}} p(\mathbf{r}_k \mid a[k], \hat{a}[k-1], \ldots, \hat{a}[k-N+2]), \quad (3.61)$$

where λ_b^{sym}, denotes the bit metric with symbol feedback and bit value b. Additional notation is for the Υ_b^{ℓ}, which is the subset of the constellation Υ that has all the symbols in the constellation with bit value b in position ℓ, e.g. assuming 4-PSK $\Upsilon_1^1 = \{0\underline{1}, 1\underline{1}\}$.

This metric can be further simplified by using bit feedback in conjunction with symbol feedback, hence the metric now becomes,

$$\lambda_b^{bit}[\ell] = \log\left(p(\mathbf{r}_k \mid \tilde{a}[k], \hat{a}[k-1], \ldots, \hat{a}[k-N+2])\right) \tag{3.62}$$

where $\tilde{a}[k] = \mathrm{X}\{\hat{\psi}[0], \ldots, \hat{\psi}[\ell-1], b, \hat{\psi}[\ell+1], \ldots, \hat{\psi}[l-1]\}$, and $\hat{\psi}[\square]$ represents the decision feedback bits.

The metrics using the ML probability metric of DFDD and substituting it for (3.61) and (3.62), now become

$$\lambda_b^{sym}[\ell] = \log \sum_{a[k]\in\Upsilon_b^{\ell}} \exp\left(2\cdot\mathrm{Re}\left\{a[k]r^*[k]\sum_{v=1}^{N-1} t_v r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j]\right\}\right) \tag{3.63}$$

and

$$\lambda_b^{bit}[\ell] = \log\left(\exp\left(2\cdot\mathrm{Re}\left\{\tilde{a}[k]r^*[k]\sum_{v=1}^{N-1} t_v r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j]\right\}\right)\right) \tag{3.64}$$

which can be simplified to

$$\lambda_b^{bit}[\ell] = \mathrm{Re}\left\{\tilde{a}[k]r^*[k]\sum_{v=1}^{N-1} t_v r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j]\right\}. \tag{3.65}$$

The DFDD decision metric is used as the soft bit metric of the inner modulation code. Both (3.64) and (3.65) are referred to as the decision feedback differential demodulation (DFDM) metrics as in [23]. As with DFDD, with an observation window $N=2$ for DFDM is equivalent to conventional coded differential detection. The bit feedback metric (3.65), is simpler than the symbol feedback metric (3.63). Both metrics do not increase exponentially with N, remaining relatively simple to compute.

3.3.3 The Iterative Decoding Algorithm

In DFDD the decision feedback symbols were determined immediately and could be used for processing the metric for the next symbol, however, this is not the case for DFDM. In DFDM there are no feedback symbols available for the first iteration, so DFDM would resort to conventional differential demodulation, i.e., with $N=2$ which as mentioned earlier would be equivalent to standard non-coherent BICM. Once the metrics are determined they are deinterleaved and used as soft inputs to the standard Viterbi decoder, which adds the respective bit metrics to constitute the branch metric. The ML sequence of bits is decoded from the Viterbi decoder. These decoded bits are re-encoded according to the channel code. This new coded bit sequence is interleaved and remodulated, and used as decision feedback bits and symbols for the next iterations.

Once there is feedback available larger observation windows may be used, i.e., $N \geq 2$. The metric used for all these iterations is (3.65), which is computationally low, however (3.63) may also be used.

As mentioned earlier from [31, 32, 49], the labelling structure affects the performance. It was determined experimentally by the authors that for all other labelling (e.g. standard and set partitioning) structures, the loss in the first iteration could not be compensated for by further iterations, and the gain in performance after the first iteration was largest with Gray labelling.

3.3.4 Convergence and Cut-off Rate Analysis

Due to the iterative nature of DFDM, the question of convergence arises. In [23], for the purposes of convergence analysis the linear prediction approach was used as a performance measure as in [21] and [22]. Instead of determining the actual estimate $r_c[k-1]$, as in [22], it was shown that the mean square error may be expressed in terms of $\rho[k]$, which is defined as

$$\rho[k] \triangleq f[k] + n[k]d^*[k], \tag{3.66}$$

and the corresponding estimate thereof was defined in [23] as

$$\hat{\rho}[k] \triangleq d^*[k-1]\sum_{v=1}^{N-1} p_v r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j], \tag{3.67}$$

where p_v are the predictor coefficients. It can be seen that from (3.67), the minimum mean square estimate will correspond to $\hat{a}[k-j] = a[k-j]$, which is the genie aided case. Hence it was appropriate to choose the mean square estimation error variance as the convergence metric. It may be defined in terms of the $\rho[k]$ as

$$\sigma_e^2 \square \mathrm{E}\left\{\left|\rho[k]-\hat{\rho}[k]\right|^2\right\}. \tag{3.68}$$

It is shown in [23], that for the genie aided curves, the error variance is lower than the standard $N=2$ case. Hence if σ_e^2 decreases from that of $N=2$ for observation intervals $N>2$ after successive iterations with normal decision feedback symbols, then the corresponding BER will likewise decrease and converge towards the genie aided case. If however, the error variance increases beyond the $N=2$ case with the decision feedback symbols for larger N, then the BER would also increase above that of the standard BICM system and hence diverge. The error variance may be re-written as

$$\sigma_e^2 \square \mathrm{E}\left\{\left|\rho[k]\right|^2\right\}-2\cdot\mathrm{Re}\left\{E\left\{\rho[k]\hat{\rho}^*[k]\right\}\right\}+\mathrm{E}\left\{\left|\hat{\rho}[k]\right|^2\right\}. \tag{3.69}$$

which simplifies to

$$\sigma_e^2=\sigma_f^2+\sigma_n^2-2\cdot\mathrm{Re}\left\{\mathrm{E}\left\{\rho[k]\hat{\rho}[k]\right\}\right\}+\mathrm{E}\left\{\left|\hat{\rho}[k]\right|^2\right\}. \tag{3.70}$$

For mathematical simplicity the authors made the assumption that the decision feedback symbols are mutually statistically independent, and independent of the corresponding channel gain, which of course in a practical system is not the case. The received sequence $r[k]$, may be re-written as

$$r[k]=d[k]f[k]+n[k]=a[k]\cdot\left(f[k]+n[k]d^*[k]\right)\cdot d[k-1], \tag{3.71}$$

using the differential relation

$$a[k]=d[k]d^*[k-1]. \tag{3.72}$$

Using (3.71), with some statistical manipulations the third and fourth terms on the right hand side of (3.70), become

$$\mathrm{E}\left\{\rho[k]\hat{\rho}[k]\right\}=\sum_{v=1}^{N-1}p^*[v]R_f[v]\prod_{j=1}^{v-1}\mathrm{E}\left\{a[k-j]\hat{a}[k-j]\right\}\text{, and} \tag{3.73}$$

$$\mathrm{E}\left\{\left|\hat{\rho}[k]\right|^2\right\}=\sum_{v=1}^{N-1}\sum_{\mu=1}^{N-1}p[v]p^*[\mu]\left(R_f[\mu-v]+\sigma_n^2\delta[\mu-v]\right)\prod_{j=\min\{v,\mu\}}^{\max\{v,\mu\}-1}\mathrm{E}\left\{a[k-j]\hat{a}^*[k-j]\right\}, \tag{3.74}$$

where $\delta[\square]$ is the Kronecker delta.

With each iteration, the error variance needs to be recalculated to determine whether it is decreasing or increasing, hence whether it is converging or diverging. However, for this to be seen the expectations were required to be a function of the BER. To calculate the expectation with respect to the bit error rate, the constellation and labelling needs to be considered. Assuming Gray labelling and 4-PSK or 8-PSK constellations, the expectation was defined as

$$\chi \,\square\, \mathrm{E}\left\{a[k-j]\hat{a}^{*}[k-j]\right\}, \tag{3.75}$$

which for 4-PSK becomes

$$\chi = 1-2\cdot\mathrm{BER}, \tag{3.76}$$

and for 8-PSK was

$$\chi = 1-(3-\sqrt{2}/2)\cdot\mathrm{BER}+(2-\sqrt{2})\cdot\mathrm{BER}^{2}, \tag{3.77}$$

where the BER is the bit error rate of the previous iteration.

Using these equations and substituting into (3.70), the error variance was determined as

$$\begin{aligned}\sigma_e^2 = \sigma_f^2 + \sigma_n^2 - 2\cdot\mathrm{Re}\left\{\sum_{v=1}^{N-1} p^{*}[v]R_f[v]\cdot\chi^{v-1}\right\} + \\ \sum_{v=1}^{N-1}\sum_{\mu=1}^{N-1} p[v]p^{*}[\mu]\left(R_f[\mu-v]+\sigma_n^2\delta[\mu-v]\right)\chi^{|v-\mu|}\end{aligned} \tag{3.78}$$

It was shown in [23] that for low SNRs the error variance increased above $N=2$ after two iterations with decision feedback symbols, while for higher SNRs the error variance converged to that of the minimum variance represented by the genie aided lines. Hence, it was determined that the DFDM scheme should converge and provide better performance for SNRs of interest. It was also determined experimentally that the majority of the performance gain is achieved after the second iteration, which is the first iteration to use larger observation windows.

Another method to confirm performance gain of the DFDM was through the cut-off rate, which is a common measure of performance for information throughput. It was determined in [31], that for the BICM system the cut-off rate is given by

$$R_0 = l \cdot \left(1 - \log_2(B+1)\right), \tag{3.79}$$

where B is the so-called average Bhattacharyya factor, defined by the bit metrics as

$$B \square \frac{1}{l}\sum_{\mu=0}^{l-1} \mathrm{E}\left\{\sqrt{\frac{\exp\left(\lambda_{\bar{b}}^{\times}[\mu]\right)}{\exp\left(\lambda_{b}^{\times}[\mu]\right)}}\right\}, \quad \times \in \{\text{sym, bit}\}, \tag{3.80}$$

for which both derived metrics may be used. For the largest computational savings the bit feedback metric was used, with genie aided feedback. It was shown in [23] that for slow fading channels, there is a definite gap between the $N = 2$ and perfect channel state information (CSI) case. The genie aided curves showed improvement in the cut-off rate when larger observation windows were used. The cut-off rate analysis also confirmed that for fast fading channels the improvement for DFDD genie aided was negligible, and hence would not be viable for use in fast fading channels. The perfect CSI and $N = 2$ are used as reference curves to confirm that the genie aided DFDM decreases the performance gap between coherent and non-coherent detection.

Using the mean square error variance, and the cut-off rate, it is seen that for relevant SNRs the DFDM system converges to the genie aided case which is the performance bound for this feedback scheme. Provided the DFDM with decision feedback symbols converges to the genie aided case then there will be a corresponding improvement in the cut-off rate, which all translates to an improvement in BER performance, which is confirmed by simulation in [23]. It should be mentioned that the authors have recently improved the DFDM structure and given a much more in-depth analysis on the new and improved iterative DFDM structure in [36].

3.4 Simulation Results

In this section the replicated simulation results are shown for the DFDD of [21], the RLS-DFDD of [22], and the DFDM of [23]. These simulation results will be used for comparison purposes. In all cases the BER curves were replicated for the stated conditions and parameters in the respective papers, unless otherwise stated. It is important to note that the simulation results are plotted with respect to the bit energy to noise, and this is consistent throughout this dissertation. The discussions may use SNR, because of the direct relationship with the bit energy. Therefore a corresponding trend will be seen for SNR and bit energy respectively.

The DFDD for flat fading channels [21], was derived directly from the conditional probability density function, and hence will be regarded as the optimal DFDD benchmark.

3.4.1 Standard DFDD Results

As derived earlier the optimal filter coefficients may be determined by the autocorrelation of the fading process and the noise variance. Assuming a normalized Doppler frequency $F_dT = 0.03$, the optimal coefficients for the fading process according to the Jakes model are $R_f[0]=1$, $R_f[1]=0.9911$, $R_f[2]=0.9648$, $R_f[3]=0.9216$ and $R_f[4]=0.8628$.

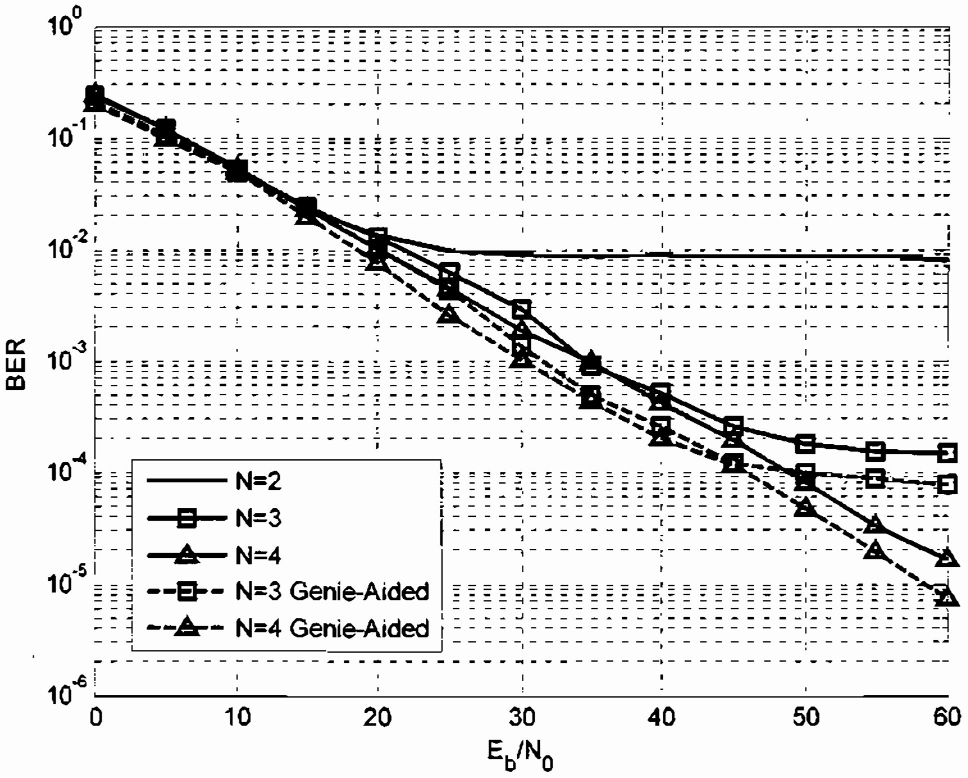

Fig. 3.4 Standard DFDD results for F_dT=0.03, for N=2,3,4, optimal coefficients used.

In Fig. 3.4 the optimal results for standard DFDD is shown, where it can be seen that the $N = 2$ curve, which is equivalent to SDD floors at just below 1E^{-2} (10^{-2}). The performance of DFDD for $N > 2$, shows significant gains over SDD. It is also noticed that there is virtually a constant gap between the genie aided DFDD and the DFDD with decision feedback symbols, which is to be expected due to erroneous feedback symbols.

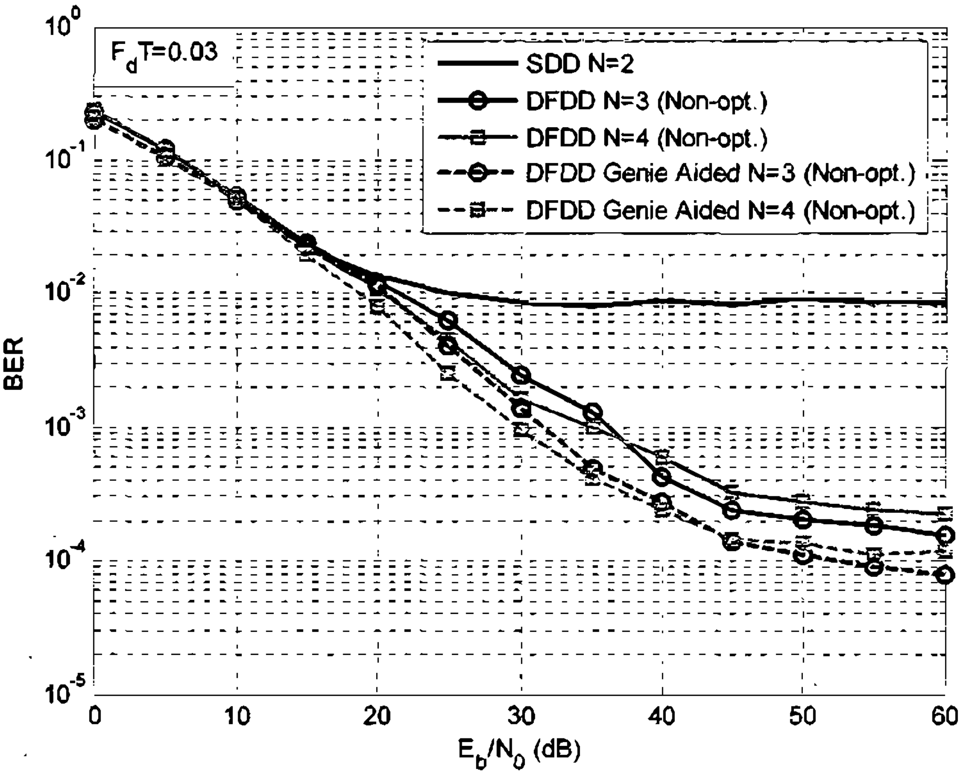

Fig. 3.5 Standard DFDD results for non-optimal filter coefficients, with F_dT=0.03.

However, it was shown in [21] that for the correlation values: $R_f[0]=1$, $R_f[1]=0.9917$, $R_f[2]=0.9671$ and $R_f[3]=0.9267$, the performance as shown in Fig. 3.5 is suboptimal with early flooring evident. These correlation values were used in [21] for the normalized bandwidth $F_dT=0.03$. The penalty for marginally suboptimal coefficients may be as high as an order of magnitude if no channel coding is applied. The results shown in Fig. 3.5 are virtually exactly what was produced in [21].

3.4.2 RLS-DFDD Results

It was shown in [22] that RLS algorithm was stable for a Rayleigh fading channel with $F_dT=0.03$. For the range of forgetting factors $0.95 \le \Lambda \le 1$ the BER remained virtually flat. Hence to maximize memory for the RLS algorithm, while still giving some tracking capability, $\Lambda=0.99$ was used for the simulation results.

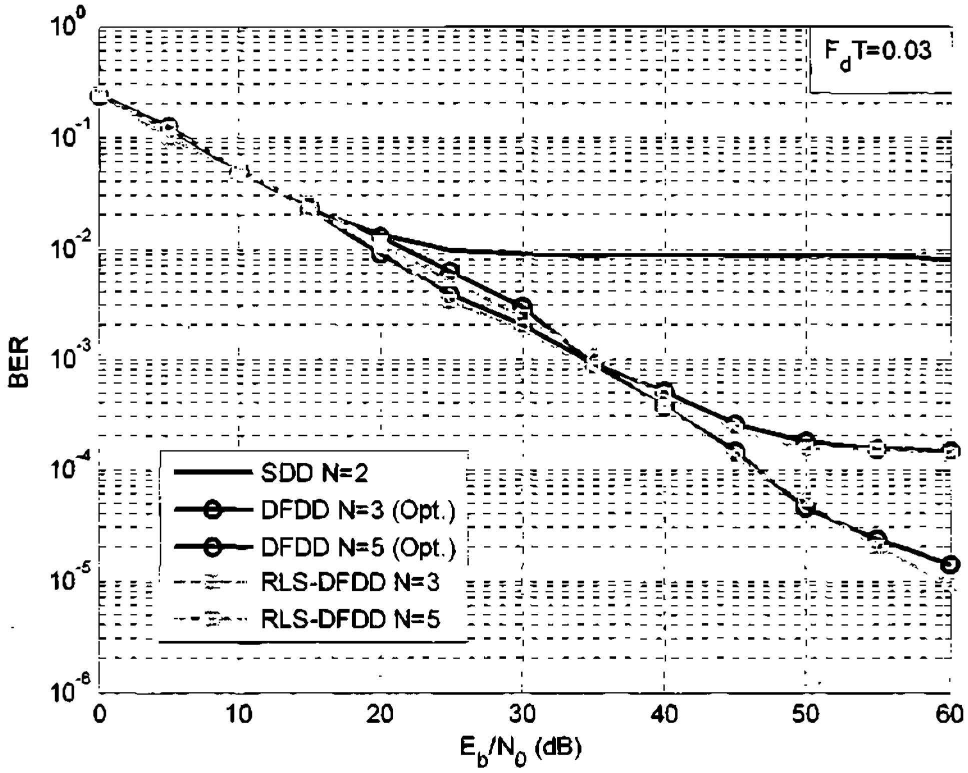

Fig. 3.6 A comparison of the DFDD vs. the RLS-DFDD. Optimal coefficients were used. F_dT=0.03.

In Fig. 3.6, a comparison is drawn between standard DFDD and RLS-DFDD. It is clear that RLS-DFDD is equivalent in performance to the DFDD of [21] with optimal coefficients. This proves that RLS-DFDD would not suffer from a performance penalty related to incorrect channel statistics as with the DFDD with fixed coefficients. It is noted that there is a similar performance pattern for genie aided DFDD and RLS-DFDD, where both curves virtually overlap each other. It should be noted that the results shown in [22], did not match the replicated curves of Fig. 3.6. However, it was shown that RLS-DFDD is virtually overlapping the DFDD curve. The $N = 5$ floored earlier than the curve shown in Fig. 3.6. This slight discrepancy is most likely due to the accuracy of the fading generator used. However, the principle is still accurately demonstrated in Fig. 3.6.

3.4.3 Iterative DFDM Results

The BER for iterative DFDM with 4-PSK is shown in Fig. 3.7, where observation windows of $N = 2,3,5$ were used. The $N = 2$ curve, and perfect CSI curve form performance bounds for the iterative DFDM scheme.

It is seen that for observation intervals $N > 2$, at low SNRs the BER is above that of the standard $N = 2$ case, while at higher SNRs the actual decision feedback converges to the genie aided case. It is also seen that there is still a performance gap between the perfect CSI case and DFDM. These curves are almost exactly those shown in [23]. These results were achieved after 100 frame errors, not just bit errors, to ensure that even with burst errors a proper average is achieved.

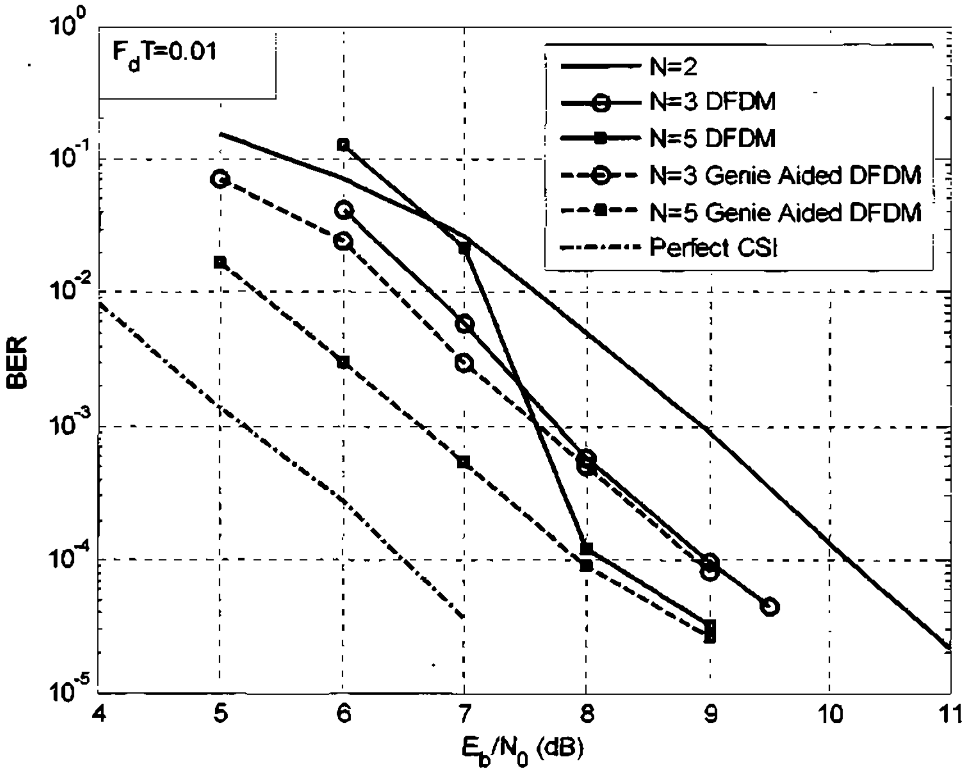

Fig. 3.7 BER results of the DFDM for observation windows of N=2, 3, 5. Associated genie aided curves and perfect CSI case also shown

3.5 Conclusion

This chapter has covered the derivation and performance of various decision feedback schemes used for MSD. The curves replicated match those in their respective publications.

The equivalence of prediction based DFDD, and the DFDD for Rayleigh flat fading channels was derived. It was seen that the adaptive RLS-DFDD offered the most attractive solution among the DFDD, the prediction based DFDD, and the adaptive RLS-DFDD for flat fading channels. It was seen that marginally suboptimum coefficients will yield early error floors, and as a worst case, an error floor above that of SDD as shown in [21]. However, it is also known that the RLS algorithm, suffers from numerical instability and RLS-DFDD may perform poorly under time selective conditions.

The DFDM offers high performance gains, for a relatively simple decision metric. Although still marginally inferior to coherent detection (perfect CSI), it offers an attractive non-coherent solution for flat fading environments, while keeping receiver complexity low, which is important for mobile communications. The use of hard decision feedback is obviously suboptimal. However, the complexity for soft decision feedback would increase, and the performance improvement may not warrant the additional power and computational requirements. If suboptimum coefficients are used, it is expected that the genie aided performance bound would be closer to the $N = 2$ curve and then even if convergence occurs, it would be to a suboptimal bound.

Chapter 4
Adaptive DFDD with the QR Decomposition

The introduction of DFDD for flat fading channels [21], was significant in improving the performance of non-coherent differential detection. The decision metric was simple and yielded results that lowered the error floor or virtually eliminated it. Although it did not require exact channel state information (CSI), the accurate second order channel characteristics were necessary for optimal performance.

The channel characteristics, for slowly time varying channels would not impact the performance of the DFDD detector significantly. For rapidly time varying channels, the characteristics would have to be updated regularly, because they may change rapidly, due to the high mobility of the channel. In this case DFDD might not be able to offer significant performance gains over standard differential detection (SDD). In this case it would be preferable to have an adaptive DFDD scheme.

An adaptive DFDD scheme was introduced in [22], which incorporated the recursive least squares (RLS) algorithm, referred to as RLS-DFDD. This provided an alternate, blind adaptive DFDD scheme which offered the same performance as the DFDD with optimal filter coefficients. This adaptive scheme would be favourable over the DFDD, if channel characteristics are not accurate. The performance penalty with marginally suboptimum coefficients was shown in the previous chapter, where early flooring was observed.

However, the performance of RLS-DFDD is also dependent on the channel characteristics remaining time invariant, otherwise the RLS algorithm may become unstable numerically, and therefore performance degrades. Other adaptive schemes were developed to improve the performance of differential detection, using different adaptive algorithms. The ACE detector of [43] used the recursive least squares lattice filter to improve the performance over standard non-coherent detection. An alternate decision feedback scheme was proposed in [44], which used the LMS algorithm but could not be used for fast fading channels and also required training. Both [43] and [44] require high filter orders for their respective performance gains. The mixture Kalman filtering approach was used in [45], which performed near the optimal performance bound for perfect feedback.

In this chapter, a new robust adaptive DFDD scheme is introduced that uses the QR decomposition, which is superior to the RLS-DFDD for Rayleigh time selective channels, and is comparable to the DFDD with optimal coefficients. In section 4.1, the least squares problem, and its theoretical solution using the weighted QR decomposition is described. The use of the QR decomposition to adaptive DFDD, is outlined in section 4.2. The generic QR decomposition based DFDD (QR-DFDD) system of section 4.2 is modified in section 4.3, to yield a sliding window based approach to QR-DFDD. The initialization for both algorithms, are discussed in section 4.4, while a summary of the sliding window approach is described in section 4.5. Simulation results and comparisons are illustrated in section 4.6, thereafter the chapter is concluded.

4.1 The QR Decomposition and the Least Squares Problem

The QR decomposition offers a numerically stable method of computing the full rank least squares problem:

$$\mathbf{Ax} \approx \mathbf{b}. \tag{4.1}$$

The straight forward solution $\hat{\mathbf{x}}$ that minimized the error $\|\mathbf{Ax}-\mathbf{b}\|$, is $\hat{\mathbf{x}} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}$. However, for cases when the matrix $\mathbf{A}$ is ill-conditioned, this method is not advisable. This is further endorsed by the concept of the condition number, which was touched on in chapter 2. The condition number is defined from [42] as:

$$\kappa(\mathbf{A}) \square \|\mathbf{A}\|\|\mathbf{A}^{-1}\|. \tag{4.2}$$

The condition number is an indication of the degree to which a matrix is ill-conditioned. An ill-conditioned matrix is a matrix which is close to being rank deficient. The problem with the straight forward solution indicated above is that $\kappa(\mathrm{A}^H\mathrm{A}) = (\kappa(\mathrm{A}))^2$, therefore the ill-conditioning effect is doubled. Another way of describing the condition number $\kappa(\square)$, is in terms of the eigenvalues of the matrix also shown in [42], where the condition number may be defined as:

$$\kappa(\mathrm{A}) = \frac{\sqrt{\lambda_{\max}^{eig}}}{\sqrt{\lambda_{\min}^{eig}}}, \tag{4.3}$$

where $\lambda_{\max}^{eig}$ and $\lambda_{\min}^{eig}$ are the largest and smallest eigenvalues of A, respectively. Therefore when the eigenvalue spread of a matrix is large, it also implies that the condition number of the matrix is large, and hence the matrix is ill-conditioned.

The primary advantage of solving least squares problems with QR algorithms is not speed, but the numerical stability in dealing with a system with eigenvalues $\left\|\lambda^{eig}\right\|$ instead of the normal equations with eigenvalues $\left\|\lambda^{eig}\right\|^2$.

The QR decomposition, which is computed using the sequence of Givens rotations was covered in chapter 2. Using the alternate form of the QR decomposition where

$$\mathbf{QA} = \mathbf{R}, \tag{4.4}$$

the least squares problem of

$$\mathbf{Ax} = \mathbf{b}, \tag{4.5}$$

will be solved, which is summarized from [41].

The solution of the least squares problem introduces an error in the estimate, especially in noisy environments. Hence the error in the estimate is denoted as

$$e(i) = d_r(i) - \hat{\mathbf{x}}^H \mathbf{u}(i), \tag{4.6}$$

where $d_r(i)$ is the desired response, $\hat{\mathbf{x}}$ is the least squares solution, and $\mathbf{u}(i)$ is the pre-windowed input samples. As in the RLS algorithm an exponentially weighted factor is introduced as a memory factor, so that in non-stationary cases the effect of earlier data is weighted less than the more recent samples. The index of performance with the weighting factor now becomes

$$\xi(n) = \sum_{i=1}^{n} \omega^{n-i} \left|e(i)\right|^2 , \tag{4.7}$$

where $n \geq L$ is the input sample length, L is the dimension of the solution, i is a time index for the sample data, $1 \leq i \leq n$, and ω is the weighting factor associated with i. These definitions are alternately expressed in a vector form. The $n \times 1$ error vector is defined as

$$\boldsymbol{\varepsilon}^H(n) \square [e(1),\ e(2),\ldots,\ e(n)], \tag{4.8}$$

while the $n \times 1$ desired response vector is defined as

$$\mathbf{b}^H(n) \square [d_r(1),\ d_r(2),\ldots.,\ d_r(n)], \tag{4.9}$$

and the $n \times n$ exponential weight matrix is defined as

$$\Lambda(n) = \text{diag}[\omega^{n-1}, \omega^{n-2}, \ldots, 1]. \tag{4.10}$$

The QR factorization decomposes $\mathbf{A}$, using the unitary matrix $\mathbf{Q}$ into an upper triangular matrix $\mathbf{R}$. Noting that the Euclidean norm of a vector does not change by the pre-multiplication of any unitary matrix, the performance index becomes

$$\xi(n) = \left\|\mathbf{Q}(n)\Lambda^{1/2}(n)\boldsymbol{\varepsilon}(n)\right\|^2 , \tag{4.11}$$

where $\mathbf{Q}(n)$ is any $n \times n$ unitary matrix. Re-writing the error vector as

$$\boldsymbol{\varepsilon}(n) = \mathbf{b}(n) - \mathbf{A}(n)\mathbf{x}(n), \tag{4.12}$$

then introducing the unitary and exponential weight matrices, (4.12) becomes

$$\mathbf{Q}(n)\Lambda^{1/2}(n)\boldsymbol{\varepsilon}(n) = \mathbf{Q}(n)\Lambda^{1/2}(n)\mathbf{b}(n) - \mathbf{Q}(n)\Lambda^{1/2}(n)\mathbf{A}(n)\mathbf{x}(n). \tag{4.13}$$

At this point $\mathbf{Q}(n)$ is an arbitrary unitary matrix. However, applying the QR decomposition to $\Lambda^{1/2}(n)\mathbf{A}(n)$, for a specific $\mathbf{Q}(n)$ it is seen that

$$\mathbf{Q}(n)\Lambda^{1/2}(n)\mathbf{A}(n) = \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{0} \end{bmatrix}, \quad n \geq L, \tag{4.14}$$

where $\mathbf{R}(n)$ is an $L\times L$ upper triangular matrix, and $\mathbf{0}$ is an $(n-L)\times L$ zero matrix. Using this relation in (4.13), the third term in the equation becomes

$$\mathbf{Q}(n)\Lambda^{1/2}(n)\mathbf{A}(n)\mathbf{x}(n)=\begin{bmatrix}\mathbf{R}(n)\\ \mathbf{0}\end{bmatrix}\mathbf{x}(n)=\begin{bmatrix}\mathbf{R}(n)\mathbf{x}(n)\\ \mathbf{0}_{n-L}\end{bmatrix}, \tag{4.15}$$

where the result is an $n\times 1$ vector, with $\mathbf{0}_{n-L}$ being a $(n-L)\times 1$ zero vector.

If the unitary matrix is partitioned into an $L\times n$ matrix $\mathbf{F}(n)$, and an $(n-L)\times n$ matrix $\mathbf{S}(n)$, representing the first L rows, and the remaining rows respectively. With this partition the first term on the right hand side of (4.13) becomes

$$\mathbf{Q}(n)\Lambda^{1/2}(n)\mathbf{b}(n)=\begin{bmatrix}\mathbf{F}(n)\\ \mathbf{S}(n)\end{bmatrix}\Lambda^{1/2}(n)\mathbf{b}(n)=\begin{bmatrix}\mathbf{g}(n)\\ \mathbf{h}(n)\end{bmatrix}, \tag{4.16}$$

where it follows that the vectors $\mathbf{g}(n)$ and $\mathbf{h}(n)$ are defined as

$$\mathbf{g}(n)=\mathbf{F}(n)\Lambda^{1/2}(n)\mathbf{b}(n),\ \ (L\times 1 \text{ vector}), \tag{4.17}$$

and

$$\mathbf{h}(n)=\mathbf{S}(n)\Lambda^{1/2}(n)\mathbf{b}(n),\ \ ((n\text{-}L)\times 1 \text{ vector}). \tag{4.18}$$

Substituting (4.16) and (4.15) into (4.13), the equation may be represented as a partitioned vector written as

$$\mathbf{Q}(n)\Lambda^{1/2}(n)\varepsilon(n)=\begin{bmatrix}\mathbf{g}(n)\\ \mathbf{h}(n)\end{bmatrix}-\begin{bmatrix}\mathbf{R}(n)\mathbf{x}(n)\\ \mathbf{0}_{n-L}\end{bmatrix}=\begin{bmatrix}\mathbf{g}(n)-\mathbf{R}(n)\mathbf{x}(n)\\ \mathbf{h}(n)\end{bmatrix}. \tag{4.19}$$

Hence to minimize the squared norm of the performance index, we determine the least solution $\hat{\mathbf{x}}(n)$ that will solve $\mathbf{g}(n)-\mathbf{R}(n)\mathbf{x}(n)=0$. Hence, the squared norm of $\mathbf{Q}(n)\Lambda^{1/2}(n)\varepsilon(n)$ will be minimized when

$$\mathbf{R}(n)\hat{\mathbf{x}}(n)=\mathbf{g}(n). \tag{4.20}$$

It also follows that the minimum value of the performance index is determined solely by $\mathbf{h}(n)$ since $\mathbf{g}(n)-\mathbf{R}(n)\hat{\mathbf{x}}(n)=0$, therefore

$$\xi(n)_{\min}=\|\mathbf{h}(n)\|^2. \tag{4.21}$$

4.2 Adaptive DFDD with the QR Decomposition

As seen from the previous section to utilize the QR decomposition we require an input data matrix $\mathbf{A}$, which will be used as data to determine the optimal filter coefficients $\mathbf{p} = \hat{\mathbf{x}}$ for the given desired response $\mathbf{b}$.

The system model is similar to that used for the DFDD scheme in [21]. An M-PSK modulator was assumed, with differential encoding applied. The Rayleigh frequency flat fading channel is time selective, with AWGN. Both the noise and fading processes are mutually uncorrelated complex random Gaussian processes. The fading process is correlated in time and modelled using the Jakes model [47]. The appropriate normalizations for the fading and symbol energy, as in [21] still applies. The notation for system and channel parameters in chapter 3 is also employed in this chapter.

It was stated in [50], that there was no universally accepted boundary for values of F_dT, which resulted in fast fading. So in this case it is assumed that for $F_dT > 0.05$, fast fading occurs.

A linear prediction filter approach is used, where the incoming data is used to make a predictive estimate of the desired response. Similar to the derivation of the RLS-DFDD algorithm for adaptive DFDD [22], a linear estimate of the received symbol based on decision feedback symbols was used, according to the relation

$$r_e[k-1] = \sum_{v=1}^{N-1} p_v r[k-v] \prod_{\mu=1}^{v-1} \hat{a}[k-\mu] = \sum_{v=1}^{N-1} p_v \hat{r}[k,v], \tag{4.22}$$

where $r_e[k-1]$ was the estimate of the previously received symbol, p_v were the filter coefficients and $\hat{a}[k-\mu]$ were the decision feedback symbols. It was also seen that $\hat{r}[k,v]$ was defined as in [22]:

$$\hat{r}[k,v] \square r[k-v] \prod_{\mu=0}^{v-1} \hat{a}[k-\mu], \quad 1 \le v \le N-1. \tag{4.23}$$

It can be seen that (4.22) is the form of an $N-1$ order linear prediction filter.

The RLS algorithm implicitly uses data from the start of the adaptive process. However, it was shown in the previous section that the data matrix $\mathbf{A}$ would explicitly contain data from the start of the adaptive process. In (4.22) it is clear that p_v will be the required solution to the least squares problem, with $\hat{r}[k,v]$ populating the rows of the data matrix $\mathbf{A}$, and the desired response being a vector of received symbols $r[k]$. The least squares problem of $\mathbf{Ax}=\mathbf{b}$ for DFDD is now represented by $\hat{\mathbf{r}}\mathbf{p}=\mathbf{r}$ defined below:

$$\begin{bmatrix} \hat{r}[1,1] & \hat{r}[1,2] & \cdots & \hat{r}[1,N-1] \\ \hat{r}[2,1] & \ddots & & \\ \vdots & & & \\ \hat{r}[n,1] & \hat{r}[n,2] & \cdots & \hat{r}[n,N-1] \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{N-1} \end{bmatrix} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[n] \end{bmatrix}, \qquad (4.24)$$

where n is the number of information symbols in the entire frame. However, this representation is not possible unless decision feedback symbols have already been determined. Hence a stacking technique is used to determine the least squares solution as the matrix is populated, which can be interpreted as

$$\begin{bmatrix} \hat{r}[1,1] & \hat{r}[1,2] & \cdots & \hat{r}[1,N-1] \\ \hat{r}[2,1] & \ddots & & \\ \vdots & & & \\ \hat{r}[k,1] & \hat{r}[k,2] & \cdots & \hat{r}[k,N-1] \end{bmatrix} \begin{bmatrix} p_1[k] \\ p_2[k] \\ \vdots \\ p_{N-1}[k] \end{bmatrix} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[k] \end{bmatrix}, \qquad (4.25)$$

where for each new time index k there is a corresponding vector $\mathbf{p}$ that is optimal in the mean square sense. In this way decisions are made immediately and can be used as decision feedback symbols. This adaptive DFDD detector using the QR decomposition is referred to as QR-DFDD. It should be noted that for large n, the performance of QR-DFDD may not be optimal when compared to recursive algorithms like the RLS because the minimum error in estimate, $\xi(n)_{\min} = \|\mathbf{h}(n)\|^2$, where $\mathbf{h}(n)$ is an $(n-N+1)\times N-1$ matrix. Only N rows are used for the solution, therefore it is expected that for smaller n, the minimum mean square error is smaller than for large n, because of large number of rows in $\mathbf{h}(n)$.

4.3 Adaptive DFDD with QR Decomposition and Sliding Window

In QR-DFDD, the memory requirement is linearly dependent on the number of symbols in the frame. Therefore, for large frame lengths the memory and computational requirements of the QR decomposition (if a non-pipelined method is used) become large, and hence time consuming. It is also noted that the forgetting factor is exponentially weighted, hence for large frame lengths, the decision is dominated by the latest data, e.g. if $\omega = 0.9$ the weighting associated for data received $20T$ before is 0.122, and after $50T$ is 0.005. Hence, for larger frame lengths (e.g. 1000 or 10000 symbols) there are redundant calculations that would not affect the resulting solution of the QR decomposition.

To reduce the memory requirements a sliding window approach is proposed to solve for the filter coefficients. This approach introduces another memory constraint ϖ, the size of the data window used to compute the least squares solution. The sliding window approach used here works out the solution for each window independently, which is suboptimal. This sliding window approach is very similar to the method in [51], where a recursive update of the basis window through downdating was used. This would be the more optimal approach, because there is a relationship between the windows. Therefore, more information would be used. This approach however, requires additional computation. Other fast sliding window QR algorithms, were discussed in [52], which may also be applied to this system, but with additional complexity.

The least square problem using the independent sliding window approach is now represented as

$$\begin{bmatrix} \hat{r}[k-\varpi+1,1] & \hat{r}[k-\varpi+1,2] & \cdots & \hat{r}[k-\varpi+1,N-1] \\ \hat{r}[k-\varpi+2,1] & \ddots & & \\ \vdots & & & \\ \hat{r}[k,1] & \hat{r}[k,2] & \cdots & \hat{r}[k,N-1] \end{bmatrix} \begin{bmatrix} \hat{p}_1[k] \\ \hat{p}_2[k] \\ \vdots \\ \hat{p}_{N-1}[k] \end{bmatrix} = \begin{bmatrix} r[k-\varpi+1] \\ r[k-\varpi+2] \\ \vdots \\ r[k] \end{bmatrix} \tag{4.26}$$

where for each time instant k a window ϖ of previous data is used to compute the filter coefficients of the $N-1$ order linear predictor. This windowed matrix will be denoted $\mathbf{A}_k^{\varpi}$, and the corresponding desired response vector $\mathbf{b}_k^{\varpi}$. This method is suboptimum but the requirements for real time application are clearly met. The RLS algorithm requires the direct calculation of the matrix inverse, hence numerical stability comes into question. The least squares problem for the windowed DFDD is now written as

$$\mathbf{A}_k^{\varpi}\mathbf{x}_k = \mathbf{b}_k^{\varpi} \rightarrow \hat{\mathbf{r}}_k^{\varpi}\mathbf{p}_k = \mathbf{r}_k^{\varpi}, \tag{4.27}$$

where the superscript ϖ is the size of the sliding window, and the row dimension of the respective matrices.

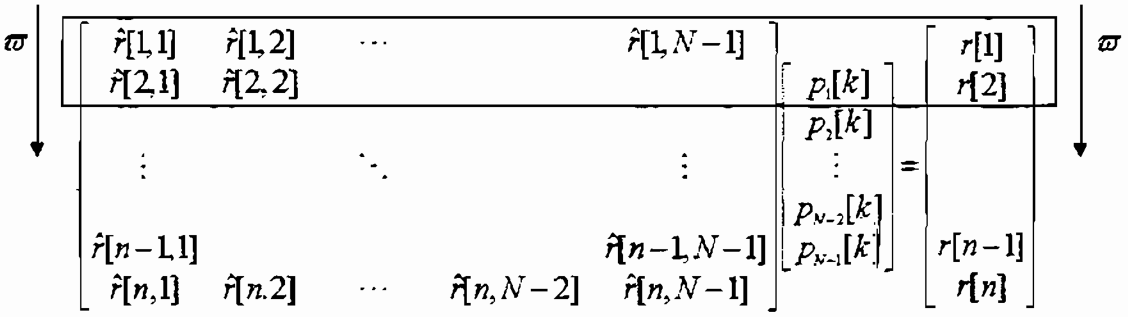
ϖ
$\hat{r}[1,1]$ $\hat{r}[1,2]$ $\cdots$ $\hat{r}[1,N-1]$
$\hat{r}[2,1]$ $\hat{r}[2,2]$
$\vdots$ $\ddots$ $\vdots$
$\hat{r}[n-1,1]$ $\hat{r}[n-1,N-1]$
$\hat{r}[n,1]$ $\hat{r}[n,2]$ $\cdots$ $\hat{r}[n,N-2]$ $\hat{r}[n,N-1]$
$p_1[k]$
$p_2[k]$
$\vdots$
$p_{N-2}[k]$
$p_{N-1}[k]$
=
$r[1]$
$r[2]$
$r[n-1]$
$r[n]$
ϖ

Fig. 4.1 Sliding window operation on the data matrix

This method of using the QR decomposition and sliding windows is referred to as QRSW-DFDD. The sliding window principle applied to DFDD is shown in Fig. 4.1, where a constant window is moving over the frame data. A special case of QRSW-DFDD is $\varpi = n-1$, then QRSW-DFDD is equivalent to QR-DFDD. It is also evident that $\varpi \geq N-1$, otherwise the least squares problem would no longer be full rank.

It is shown [41], that the QR decomposition and the least squares solution can be implemented in a pipelined fashion using systolic arrays. Another improvement that can be made for high speed real time applications is the use of CORDIC (***CO****ordinate* ***R****otation* ***DI****gital* Computation) rotations to compute the Givens rotations, discussed briefly in [42]. CORDIC rotations are a set of micro rotations based on predefined angles, which are a function of the power two. Generally any angular rotation may be represented by these micro rotations to some degree of accuracy. The advantage of the

CORDIC rotations is that it lends itself to a pipelined implementation, which is favourable to VLSI hardware implementation.

It is also worth mentioning that the exponential weight matrix for this application need not be exponential. It may be represented by fixed constants or some other function, which may prove useful for application in non-stationary environments. The application of the QR decomposition may also perform better in noisy environments since as seen in the computation the history of the data is explicitly used.

In terms of convergence the algorithm is windowed, and computes the least squares solution according to the windowed data. Once the windowed data matrix is populated then the error norm would already be optimal. It should be mentioned that there is a QR based recursive least squares algorithm, outlined in [41]. However, this alternate approach with suboptimal sliding windows is simpler.

4.4 Initialization of the QR based Algorithms

According to the definitions of the data matrix $\mathbf{A} = \hat{\mathbf{r}}$, the first $N-1$ rows are defined in terms of $\hat{r}[k,v]$. However, in reality this cannot be, because there are insufficient feedback symbols to use. Therefore the data matrix will have to be initialized until the data matrix is full rank.

In a differentially encoded transmission scheme the first transmitted symbol is always known and does not contain any information, hence with this in mind the least square problem actually does not start from $k=1$, since there is no feedback available for DFDD. Until the required number of decision feedback symbols is available conventional differential detection is used. While the decisions are being made using the conventional method, the data matrix may be populated according to the definition of $\hat{r}[k,v]$, but the feedback window used for calculating $\hat{r}[k,v]$ will change until the actual intended feedback window is reached. The initialization period is dependent on the order of the filter used, which is also the observation interval. The data matrix and desired response vector is ordered as shown

$$\begin{bmatrix} \hat{r}[2,1] & 0 & \cdots & 0 \\ \hat{r}[3,1] & \hat{r}[3,2] & \ddots & \\ \vdots & & \ddots & 0 \\ \hat{r}[N,1] & \hat{r}[N,2] & \cdots & \hat{r}[N,N-1] \end{bmatrix} \mathbf{p} = \begin{bmatrix} r[2] \\ r[3] \\ \vdots \\ r[N] \end{bmatrix}. \qquad (4.28)$$

This ensures that the QR decomposition may be applied, and it can be seen that the initial matrix is lower triangular. Since the estimation is actually for $r_e[k-1]$, there will be decision feedback symbols available for computation of $\hat{r}[k,v]$.

4.5 Summary of QRSW-DFDD

- Initialize the algorithm according to (4.28)
- Apply QR decomposition on $\mathbf{A}_k^{\varpi}$ for each k until window size ϖ is reached, then drop oldest row and shift newest data in.
- Compute least square solution $\hat{\mathbf{x}}$ according to
  - ○ $\hat{\mathbf{x}}_k = \mathbf{R}_k \backslash \mathbf{g}(\mathrm{n})$
  - ○ $\boldsymbol{\varepsilon}_k = \mathbf{b}_k - \mathbf{A}_{\varpi}^{k}\hat{\mathbf{x}}_k$, the residual error
  - ○ $e = R \backslash \mathbf{F}_k \Lambda_k^{1/2} \boldsymbol{\varepsilon}_k$, correction factor based on residual
  - ○ $\mathbf{x}_k = \mathbf{x}_k + e$
- Computing the least square solution without explicitly computing $\mathbf{Q}_k$, using only $\mathbf{R}_k$
  - ○ $\hat{\mathbf{x}}_k = \mathbf{R}_k \backslash (\mathbf{R}_k{}^H \backslash (\mathbf{A}_k^{\varpi H}\mathbf{b}_k))$
  - ○ $\boldsymbol{\varepsilon}_k = \mathbf{b}_k - \mathbf{A}_k^{\varpi}\hat{\mathbf{x}}_k$
  - ○ $e = \mathbf{R}_k \backslash (\mathbf{R}_k{}^H \backslash (\mathbf{A}_k^{\varpi H}\boldsymbol{\varepsilon}_k))$
  - ○ $\mathbf{x}_k = \mathbf{x}_k + e$

where $\mathbf{B} \backslash \mathbf{C}$ is the left matrix divide operation between $\mathbf{B}$ and $\mathbf{C}$. It is equivalent to $\mathbf{B}^{-1}\mathbf{C}$.

4.6 Simulation Results

A 4-PSK constellation is assumed for all results shown. There are two main performance factors, the forgetting factor Λ_k^{ϖ} and the sliding data window size ϖ for the QRSW-DFDD. In conjunction to these factors, the observation window size N, which dictates the order of the filter used.

The channels are modelled as Rayleigh flat fading channels, with the Jakes model used. Genie aided results are the results obtained when perfect feedback symbols are used. The frame and window sizes are stated for each subsection, as well as the respective forgetting factors.

4.6.1 Effect of Sliding Window Size

The results in this section show the effect that the sliding window has on the performance of QRSW-DFDD. QR-DFDD as mentioned is a special case of the QRSW-DFDD with the sliding window size equal to the length of the frame. The results displayed use observation windows of $N = 3$ and $N = 5$, which is sufficient to demonstrate the effectiveness of the QRSW-DFDD algorithm. The results shown in this section use a forgetting factor of unity, hence the entire window of data was weighted with 1. The frame consisted of 1000 channel symbols, or equivalently 2000 information bits.

In Fig. 4.2, the results are shown for observation window $N = 3$, with the $N = 2$, and genie aided curves shown as performance bounds for $\varpi = N = 3$ and $\varpi = 20$. The results of the genie aided DFDD with optimal coefficients were also shown for comparison purposes. In Fig. 4.2, the performance of the QRSW-DFDD was almost equivalent with the DFDD genie aided bound with optimum coefficients. At high SNRs where the fading dominated the performance, the error floor of QRSW-DFDD is lower than the DFDD.

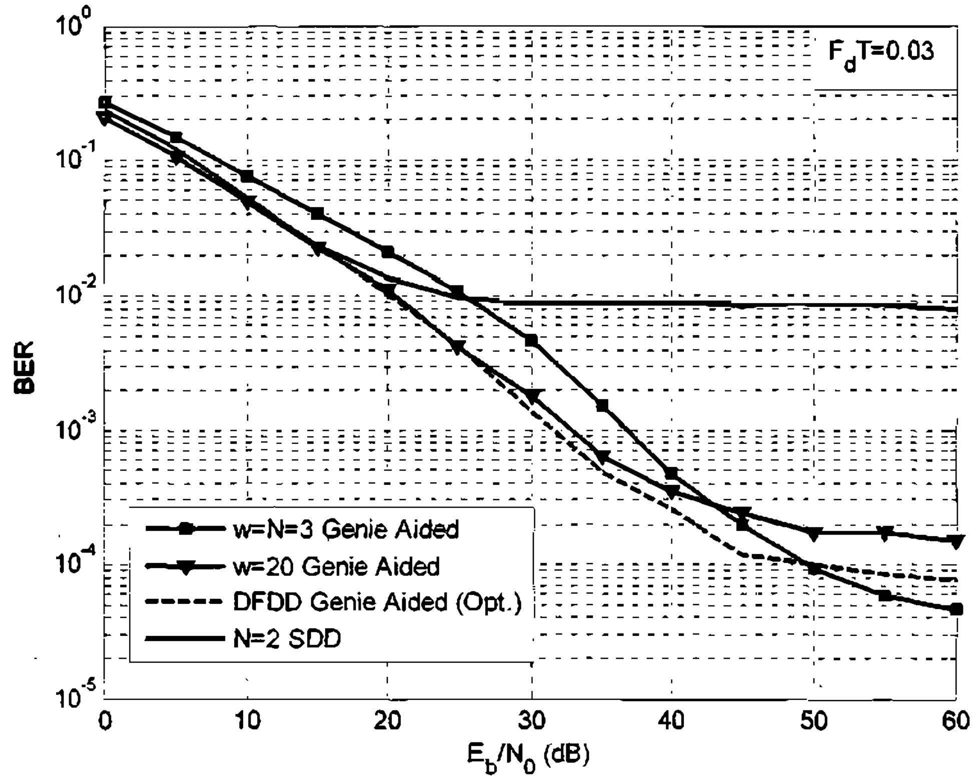

Fig. 4.2 Plot of the QRSW-DFDD genie aided bounds. Sliding window sizes of 3 and 20, and observation window size N=3. DFDD genie aided bound and SDD also shown.

The performance of QRSW-DFDD with normal decision feedback is shown in Fig. 4.3. Erroneous feedback for the QRSW-DFDD severely hinders the performance as seen for the $\varpi = N = 3$ curve where in noisy conditions the QRSW-DFDD could not adapt. However, as ϖ increases the noise averages out and the performance improves.

The QR-DFDD case clearly illustrates that at low SNRs the performance virtually matches that of the DFDD. However, at high SNRs, the tracking performance is suboptimal, depicted by the early error floor. This is due to a higher minimum error ($\xi(n)_{\min}$) with large windows.

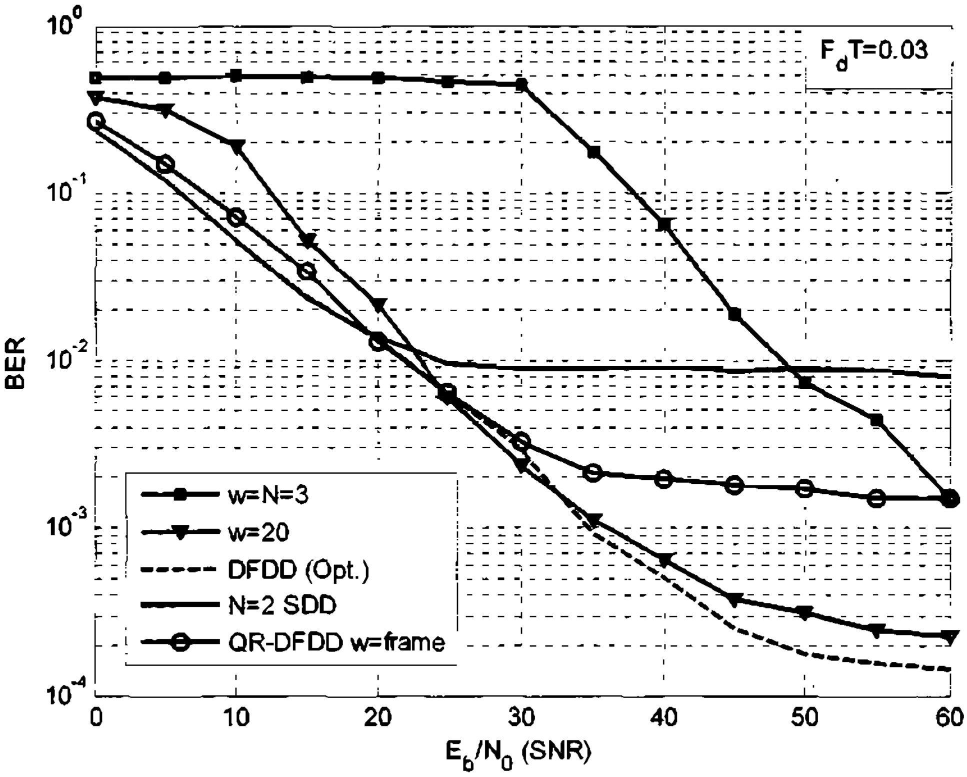

Fig. 4.3 Plot of the QRSW-DFDD and the DFDD with decision feedback. Sliding window sizes of 3 and 20, and observation window size N=3. SDD also shown.

The $\varpi = 20$ curve nearly matched the DFDD error floor. However, it does not manage to completely counteract the effect of noise at the low SNRs. The QRSW-DFDD had superior performance than SDD at low noise levels, where the fading was the dominant process. The results in Fig. 4.3 suggest that QRSW-DFDD is heavily dependent on correct feedback, since the genie aided curves of Fig. 4.2 showed near equivalent performance to the DFDD.

In Fig. 4.4, the genie aided bounds for observation window $N = 5$ are plotted. The genie aided performance bounds improve as the sliding window size increases. The QRSW-DFDD bounds approach that of the DFDD genie aided bound, but is suboptimal. However, it is clear that the performance potential is superior to that of SDD.

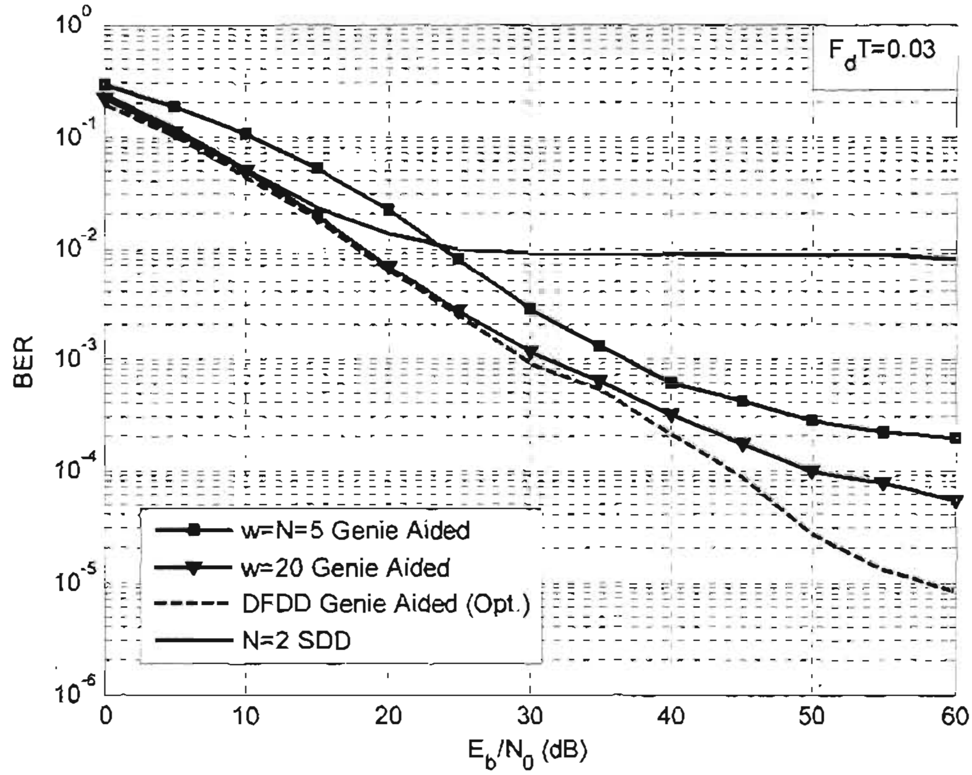

Fig. 4.4 Plot of the genie aided QRSW-DFDD bounds for sliding window sizes of 5 and 20. SDD and the genie aided DFDD are used as performance benchmarks.

A similar trend for the decision feedback curves of $N = 5$, shown in Fig. 4.5, is seen. The QRSW-DFDD is heavily reliant on correct decision feedback symbols, where in the case of $\varpi = N = 5$, the performance was worse than SDD. However, as ϖ increased, the performance improved and became superior to SDD, while the DFDD with optimal filter coefficients maintained the best performance with decision feedback. It is evident that choosing the sliding window size is dependent on the SNR, where at low SNRs a higher size is preferable; while at the higher SNRs smaller sliding windows may be used.

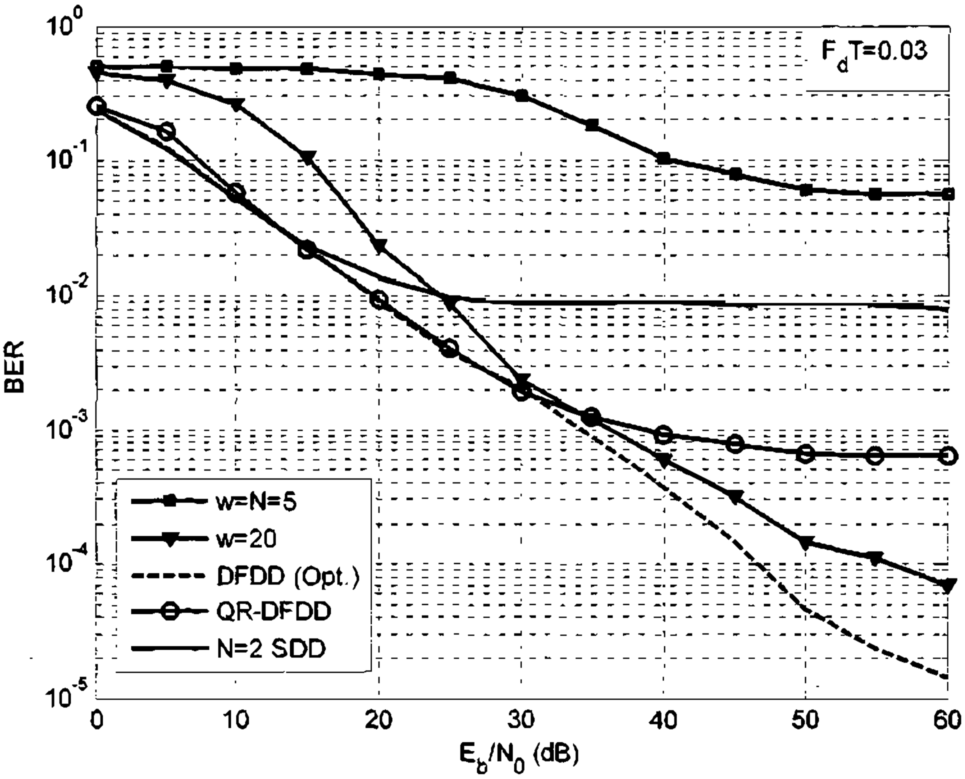

Fig. 4.5 Plot of the QRSW-DFDD for sliding window sizes of 5 and 20, for *N*=5. The SDD and the DFDD is shown as performance bounds.

The DFDD and RLS-DFDD curves were shown to be equivalent in the previous chapter for this fading bandwidth, therefore the RLS-DFDD curves were not used for comparison. In this slowly time varying, flat fading condition, the RLS-DFDD would therefore be superior to QRSW-DFDD.

4.6.2 Effect of Forgetting Factor

Although from the definition the forgetting factor matrix is made up of elements that are exponentially weighted, it may also be defined using constants. In QRSW-DFDD, due to the windowed nature of the algorithm, the forgetting factor should be weighted as high as possible, because the decision is based solely on the window, especially if the window is small with respect to the frame size. The forgetting factor matrix with constants may be written as

$$\Lambda_k^{\varpi}(0.9) = \mathrm{diag}\left\{\left[0.9, 0.9, \ldots, 0.9, 1\right]\right\}, \tag{4.29}$$

where the latest data were weighted the most with unity. In Fig. 4.6 the impact of the forgetting factor matrix was shown for constant matrices, while in Fig. 4.7 the effect of an exponential weight matrix is displayed.

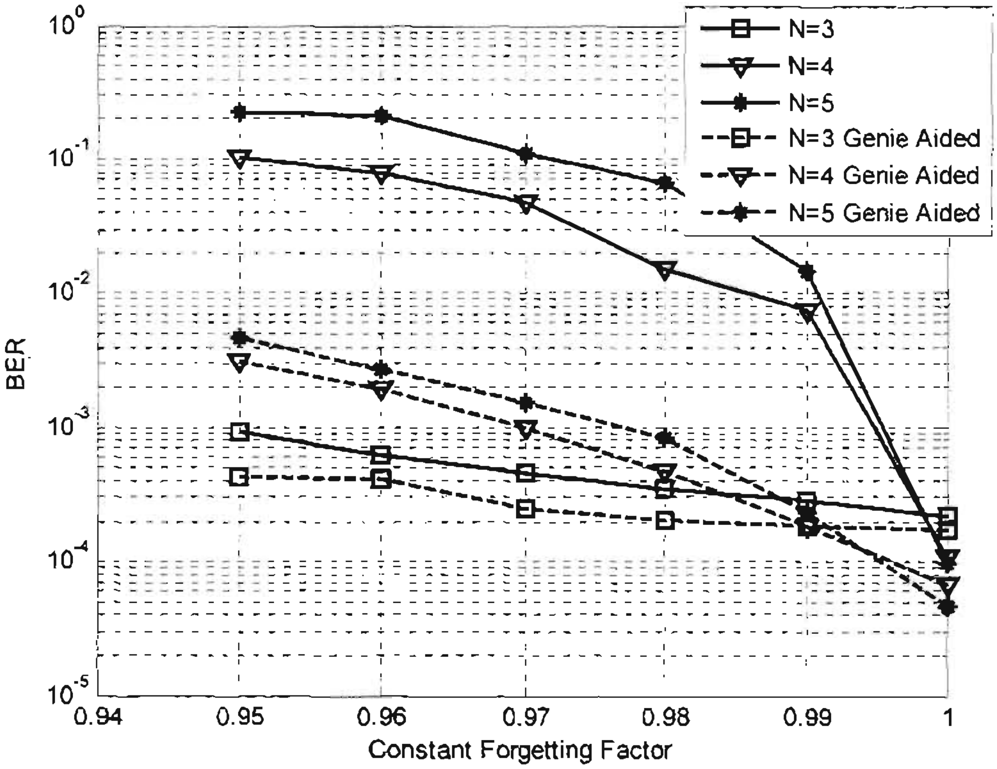

Fig. 4.6 Plot of BER vs. constant forgetting factor at 60dB, sliding window of 20. The dashed lines are for the genie aided cases.

It can be seen in Fig. 4.6 that for all observation windows, with constant forgetting factor the performance degrades. Hence it would be advisable for the QRSW-DFDD to maximize the entire possible memory of the data. The same could be said for the standard exponential forgetting factor matrix, only the performance degraded more sharply, as can be seen in Fig. 4.7. For these results a window $\varpi = 20$ was used, and a normalized Doppler frequency $F_d T = 0.03$. It should also be noted that similar trends are seen for lower SNRs, where the BER will degrade to about 0.5 as well.

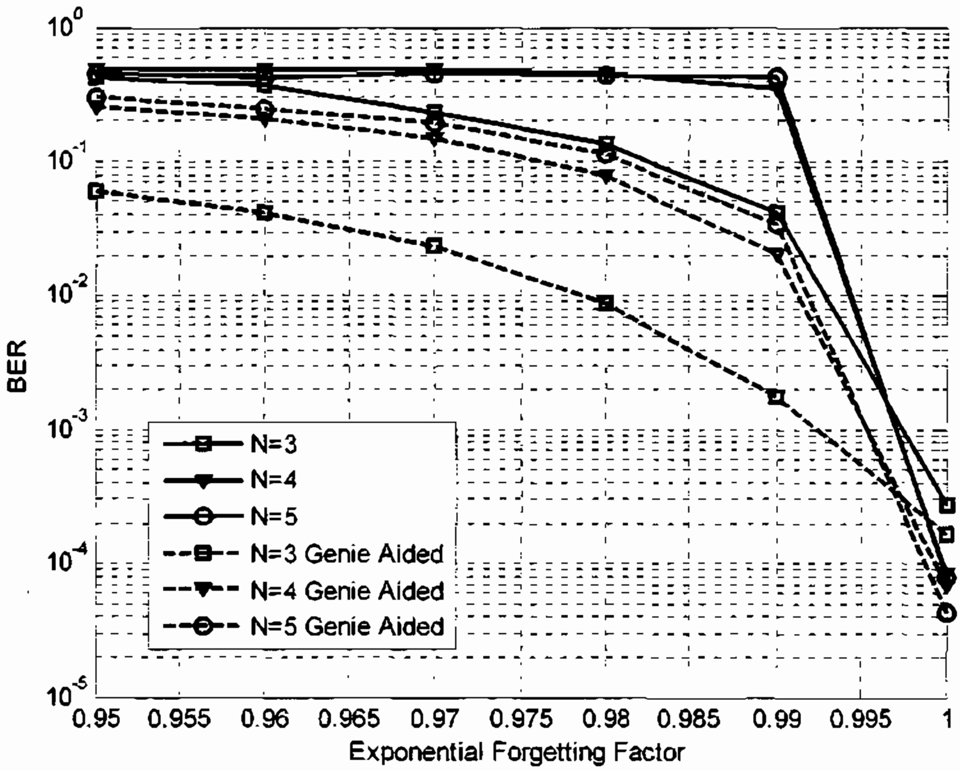

Fig. 4.7 Plot of BER vs. Exponential forgetting factor at 60dB, sliding window of 20. The dashed lines are for the genie aided cases.

4.6.3 Time Selective Fading Conditions

The reason for introducing the QR decomposition was because it should theoretically be more stable in more dynamic environments. Hence in this section fast fading conditions are used, and the corresponding performance of the DFDD, RLS-DFDD and QRSW-DFDD are shown.

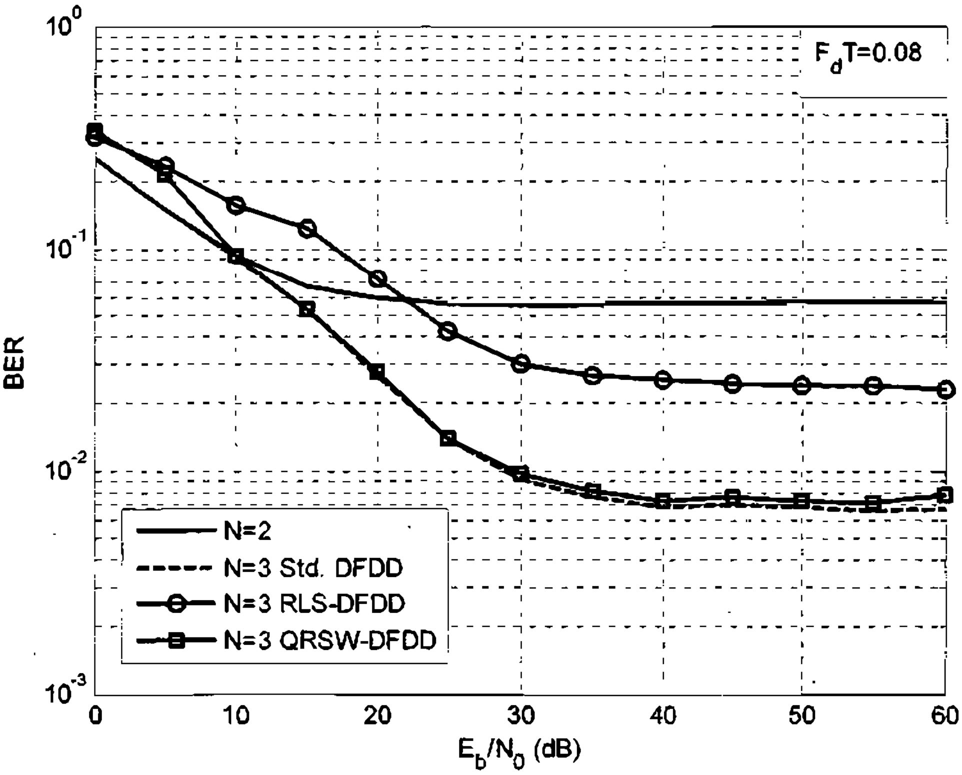

Fig. 4.8 Comparative plot of RLS-DFDD, QRSW-DFDD and DFDD with optimum metrics, for rapid flat fading channel, F_dT=0.08 and N=3.

In chapter 3, it was suspected that RLS-DFDD may have unstable and poor performance in rapid fading channels. The results for a normalized Doppler frequency of $F_dT = 0.08$ with $N = 3$, are shown in Fig. 4.8, where the DFDD, the RLS-DFDD and the QRSW-DFDD results for a rapid fading channel are compared. In Fig. 4.9, the results with an observation window $N = 4$ are plotted.

The simulation results for RLS-DFDD confirm that in rapid fading channels the RLS-DFDD does not match the DFDD with the optimum metric. It is seen, however, that the QRSW-DFDD with a sliding window size, $\varpi = 20$, was nearly equivalent to the DFDD with the optimum metric. In a fast fading environment the channel statistics, would be changing, and the accuracy of the statistics will be suboptimum, therefore the DFDD error floor would be encountered earlier. The performance gap between the DFDD and the QRSW-DFDD is small compared to that of the DFDD and the RLS-DFDD. It is therefore favourable to use the QRSW-DFDD in rapid fading environments.

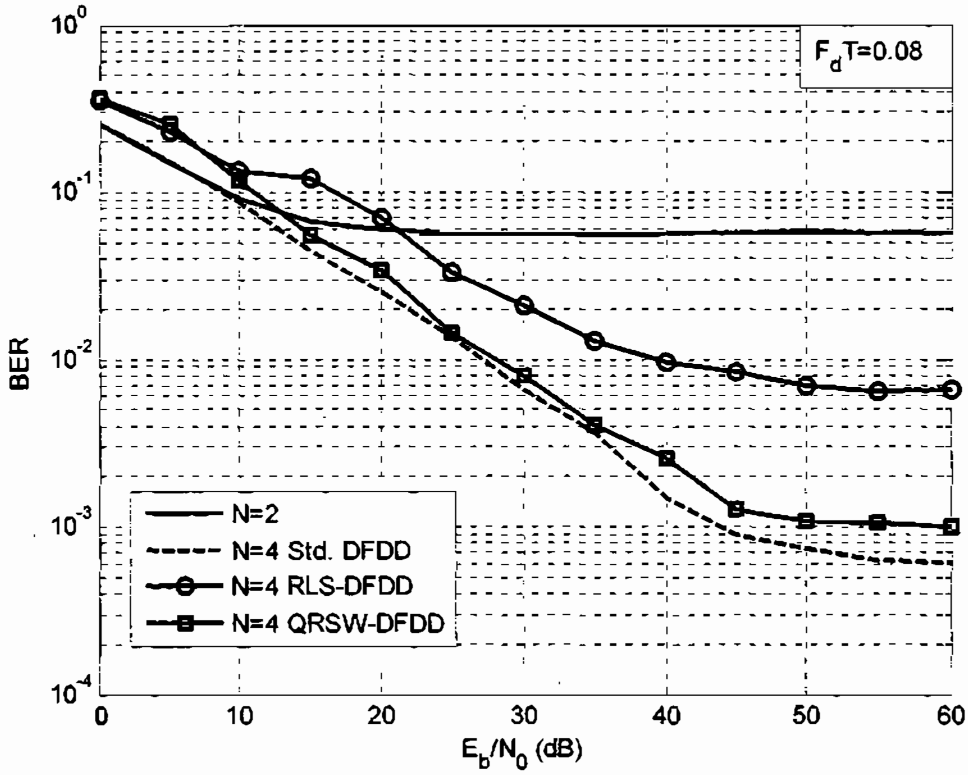

Fig. 4.9 Comparative plot of RLS-DFDD, QRSW-DFDD and DFDD with optimum metrics, for rapid flat fading channel, F_dT=0.08 and N=4.

4.7 Conclusion

The proposed QRSW-DFDD is marginally inferior to the RLS-DFDD and standard DFDD detectors under flat fading conditions. However, QRSW-DFDD showed better tracking capabilities at higher Doppler frequencies than its adaptive RLS-DFDD counterpart, which is evident from the BER performance curves, for multiple observation windows. The results under time selective conditions clearly demonstrate that the RLS-DFDD should not be used, while the DFDD due to the large Doppler frequencies may not be reliable, hence QRSW-DFDD offers an attractive solution. It is pipelined and does need large memory, while offering a stable performance under both time selective and time non-selective conditions.

The choice of sliding window size is dependent on the noise present in the channel. The forgetting factor impacts the QRSW-DFDD significantly. If the sliding window is large then a non-unity forgetting matrix may be used. However, when the window is small, which would be the most likely case for implementation, it would be more effective to use all the data available and have no forgetting factor.

The rate of convergence of the QRSW-DFDD is dependent on the sliding window size ϖ. The least squares solution is reached soon after initialization period for small sliding windows, while for RLS-DFDD, the optimal Wiener solution is approached after training, generally about 50 iterations as shown in [22]. This is also the reason that QRSW-DFDD experiences the performance loss compared to the RLS-DFDD, because the misadjustment is not necessarily small for QRSW-DFDD.

If ill-conditioned data is used then the RLS-DFDD suffers in performance as illustrated by the high frequency curves. The QRSW-DFDD was less susceptible to ill-conditioning due to the orthogonal decomposition inherent in the QR decomposition.

The computational complexity of RLS-DFDD is comparable to that of QRSW-DFDD, since for each iteration the inverse matrix had to be calculated, while for the QRSW-DFDD a series of Givens rotations are applied, dependent on the data window ϖ. However, the QRSW-DFDD approach is modular and may be implemented in a pipelined fashion for high speed real time processing, which is not the case for RLS-DFDD. The QRSW-DFDD offers performance comparable to the DFDD under flat fading conditions.

Hence the detection scheme chosen is an engineering trade-off among these various factors. It also depends on channel reliability. In a typical mobile communications system the channel statistics would have to be updated regularly, hence it may be more viable to implement the adaptive schemes described. In choosing an adaptive DFDD scheme Table 4.1, which summarizes the properties of the algorithms, may be used. In this way the appropriate trade-off may be made.

Table 4.1 Summary of Adaptive DFDD Algorithms

Property\Detector	RLS-DFDD	QRSW-DFDD
Rate of Convergence	Rapid, approaches Wiener solution	Solution in least squares form from the start.
Misadjustment	Generally small depending on channel	Does not approach Wiener solution, hence suboptimal
Robustness/Immunity	Susceptible to ill-conditioned data	More stable than RLS
Structure	Not highly modular, not suited to VLSI	Highly modular and pipelined, suitable for VLSI.
Optimal Channel Conditions	Time non-selective frequency flat fading	Time selective frequency flat fading

Chapter 5
Adaptive DFDM with RLS Algorithm

The iterative decision feedback differential modulation (DFDM) algorithm of [23], exploited the gains of decision feedback differential detection (DFDD) in Rayleigh flat fading channels [21]. It was analogously used in the iterative BICM system of the DFDM as the inner modulation code.

Other coded MSD systems were introduced in chapter 2. The DFDM was one of the simplest techniques using MSD to address the problems associated with flat fading channels. Non-coherent sequence detection [34], did not employ interleaving, which is crucial to mitigating the effects of fading. Hence its application to a fading channel would prove less effective. It is evident that the use of hard decision feedback is suboptimal, but the computational complexity associated with the SISO algorithms of [15], and [37] are extremely high.

The low complexity iterative DFDM offers many advantages over these non-coherent schemes. The soft metrics are computed using simple expressions and the presence of the convolutional code increases the error correcting capability of the scheme. The DFDM however, is reliant on the channel statistics being known, which in practical mobile systems may be inaccurate. In these cases they have to be sampled and updated regularly. The accuracy of the channel statistics will affect the soft metrics of the inner modulation code adversely as shown in [21]. Therefore, the performance gains over the conventional differential demodulation may be suboptimal. An additional drawback of the scheme was that at low SNRs, the system was unstable and caused the performance to degrade beyond that of conventional non-coherent BICM even with optimal filter coefficients.

It was seen that an alternate adaptive DFDD scheme with linear prediction equivalent to the DFDD of [21], was introduced in [22]. This adaptive scheme utilized the RLS algorithm. The adaptive DFDD algorithm is introduced to the DFDM to create a blind adaptive iterative DFDM scheme. The performance of this adaptive DFDM should be similar to that of the DFDM, because the performance of their uncoded counterparts was shown to be equivalent in Rayleigh flat fading channels in [22]. The mathematical equivalence between the linear prediction scheme DFDD and the DFDD, was shown in chapter 3 of this dissertation. The new adaptive DFDM would not require any channel information, which would be advantageous for mobile communications, where channel conditions are never constant.

This chapter is organized as follows. In section 5.1 the derivation of the metric is shown. The training routine and iterative decoding procedure is described in section 5.2. The issue of convergence for adaptive DFDM is addressed in section 5.3. An alternate form of the metric is shown in section 5.4. The simulation parameters are given in section 5.5, with the simulation results displayed in section 5.6. Conclusions are drawn in the final section.

5.1 Adaptive DFDM metric

It should be noted that the discrete time system model and symbol notations used in 3.3 of this dissertation are used. This channel model is based on the channel model described in [23], the notations however, are different.

Adopting the RLS-DFDD strategy of incorporating decision feedback symbols in the decision for calculating the predictor coefficients adaptively is used again. However, as mentioned the RLS algorithm has convergence problems under noisy conditions, and the convergence may be comparable to the LMS algorithm. It is expected that in the worst case, the performance should be worse than the DFDM, but remaining better than conventional demodulation.

The DFDM soft bit metrics may be analogously defined in terms of prediction based DFDD [22], where the symbol feedback metric is

$$\lambda_b^{sym}[\ell] \Box \log \sum_{a[k]\in\Upsilon_b^\ell} \exp\left\{2\cdot\mathrm{Re}\left\{a[k]r^*[k]\sum_{v=1}^{N-1} p_v r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j]\right]\right\}, \tag{5.1}$$

and the bit feedback metric is

$$\lambda_b^{b''}[\ell] = \mathrm{Re}\left\{\tilde{a}[k]r^*[k]\sum_{v=1}^{N-1} p_v r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j]\right\}, \tag{5.2}$$

where p_v is the normalized optimum filter coefficients. These metrics will be referred to as the adaptive RLS-DFDM metrics.

The RLS equations used in RLS-DFDD are defined as:

$$\mathbf{k}_k = \frac{\Lambda^{-1}\mathbf{J}_{k-1}\hat{\mathbf{r}}_k}{1+\Lambda^{-1}\hat{\mathbf{r}}_k{}^H\mathbf{J}_{k-1}\hat{\mathbf{r}}_k}, \tag{5.3}$$

$$\alpha[k] = r[k] - \mathbf{p}_{k-1}^T\hat{\mathbf{r}}_k, \tag{5.4}$$

$$\mathbf{J}_k = \Lambda^{-1}\mathbf{J}_{k-1} - \Lambda^{-1}\mathbf{k}_k\hat{\mathbf{r}}_k^H\mathbf{J}_{k-1}, \tag{5.5}$$

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{k}_k^*\alpha[k], \tag{5.6}$$

where the following parameters are defined as

$$\hat{\mathbf{r}}_k \Box \left[\hat{r}[k,1], \hat{r}[k,1], \ldots, \hat{r}[k,N-1]\right]^T, \tag{5.7}$$

$$\mathbf{p}_k \Box \left[p_1[k], p_2[k], \ldots, p_{N-1}[k]\right]^T, \tag{5.8}$$

$$\hat{r}[k,v] \Box r[k-v]\prod_{j=1}^{v-1}\hat{a}[k-j]. \tag{5.9}$$

The same initialization of $\mathbf{J}_0 = \Omega^{-1}\mathbf{I}$ and $\mathbf{p}_0 = [1,0,\ldots,0]^T$, as described in [22], is used. These will be used in the adaptive DFDM as well. Alternate adaptive techniques have been described in chapter 2, but due to the performance equivalence of the RLS-DFDD and the DFDD, a performance comparison can easily be shown.

5.2 The Iterative Decoding Procedure

It may be thought that since the DFDD and the RLS-DFDD are equivalent, consecutive iterations in the DFDM structure may use the RLS-DFDM metrics. An alternate simpler strategy is proposed that uses the convergence property of the RLS algorithm. As in the case of the DFDM, the first iteration would entail the use of conventional BICM detection. The assumption was made that no channel knowledge is available, hence the DFDM detector with $N=2$, may be used with an arbitrary constant filter coefficient (only 1 coefficient in this case). The filter coefficients may also be estimated using inaccurate channel characteristics that may be available. The resulting decisions from the Viterbi decoder are feedback for the subsequent training iteration.

The training iteration utilizes the RLS equations (5.3) to (5.6) for the specified observation interval N. This training iteration has a dual purpose, the metrics for the next stage of decoding are being calculated, while the optimum filter coefficients p_v, are simultaneously approached. At the end of the training iteration, p_v should be the optimal coefficients for the frame of data. The ensuing iterations will use these coefficients p_v as the filter coefficients for observation windows $N>2$. In this way the RLS only needs to be utilized once per frame, decreasing the computational complexity, required by continuous use of the RLS algorithm.

It is seen for the genie aided case the training iteration is also required. The steady state coefficients obtained after training should theoretically be close to the optimal Wiener solution. Using the steady state filter coefficients an additional iteration is required to yield the genie aided performance bound. Although perfect feedback was used for the training as well as the decisions made during the training iteration, the training procedure might take long to converge. This implies that full performance gain is not achieved after just the training iteration. Therefore, an additional iteration using these optimum coefficients, in conjunction with perfect feedback should yield the optimum performance bound. The iterative procedure is summarized in Table 5.1, where the genie aided case will only reach the third iteration.

Table 5.1 Summary of Iterative Decoding Scheme for Adaptive DFDM

Iteration #	Role in Adaptive DFDM
1	$N=2$, Standard differential demodulation, determines decision feedback for next iteration.
2	$N>2$, Training for observation window N, to estimate optimal coefficients p_v, $1 \le v \le N-1$ for frame.
3 →END	Use p_v as fixed predictor coefficients for remaining iterations

5.3 Convergence of Adaptive DFDM

The approach used in [23] to illustrate convergence as discussed earlier was to use a linear predictor of the two random processes, the fading and the noise respectively. It was shown in [22] that statistically the optimal solution for the linear predictor approach is the Wiener solution, based on the autocorrelation matrix and noise variance. It was also shown in [22] that the RLS-DFDD detector filter coefficients converge toward the optimal Wiener solution. Provided during the training iteration the optimal Wiener solution is approached then the adaptive DFDM scheme with decision feedback would also converge to the genie aided performance bound, just as the DFDM.

5.4 Alternate Approach to Adaptive DFDM

As derived earlier the prediction based DFDD metric is equivalent to the standard DFDD metric. The prediction based metric is the normalized version of the standard DFDD metric, where the normalizing factor is the mean of the error variance. The random noise and fading processes are weakly stationary, then provided the frame is large; the average mean square error variance of the adaptation process may be approximated by the mean square error variance of the frame during the training iteration, where

$$\tilde{\sigma}_e^2 = \mathrm{E}\left\{\left|\alpha[k]\right|^2\right\},\ \forall k, \qquad (5.10)$$

hence the metric now takes the form of the standard DFDM metric, with

$$\hat{t}_v = \frac{p_v}{\tilde{\sigma}_e^2}. \tag{5.11}$$

It is seen that for the bit feedback metric λ_b^{bit}, there should be no performance difference between the normalized metric and the standard metric, because they are mathematically equivalent. The symbol feedback metric λ_b^{sym} on the other hand might differ since the metric is exponentially weighted and added. However, it was mentioned in [23], that the cut-off rates associated with λ_b^{sym}, was marginally inferior to that of λ_b^{bit}, so either method should have negligible performance difference.

5.5 Simulation Parameters and Models

The standard 4-PSK constellation is used, with Gray labelling employed. The BER is displayed as a function of the average bit energy-to-noise ratio (E_b/N_0). The convolutional code used in [23] for the 4-PSK case was a 64 state, half rate convolutional code with memory 6 and generator polynomial $(133, 171)_8$. Standard Viterbi decoding was applied. For comparison purposes the same conditions are used. It was not mentioned in [23], whether zero forcing the trellis to the zero state was employed, but it was mentioned in [15], that there is negligible performance difference with or without zero state forcing. In this case zero forcing will be used, hence additional bit redundancy is added to the information bit frame before convolutional coding.

Random bit interleaving is assumed, for 4000 bits, hence there are 2000 channel symbols. The random interleaving introduces transmission diversity. The Jakes fading model is assumed with normalized Doppler frequency of 0.01. The forgetting factor of the RLS algorithm was 0.99. It is worth mentioning that the fading generator[‡] used, could drastically affect the resulting BER, especially if trying to replicate the results of published work.

[‡] The actual fading generator used is described in the appendix.

5.6 Simulation Results

The adaptive DFDM yields better results than the standard DFDM as depicted by the two plots Fig. 5.1 and Fig. 5.2, where the genie aided case and normal decision feedback cases are investigated, respectively. The curves for the genie aided adaptive DFDM are achieved after 1 training iteration as described earlier, while the normal decision feedback results are achieved after 5 iterations.

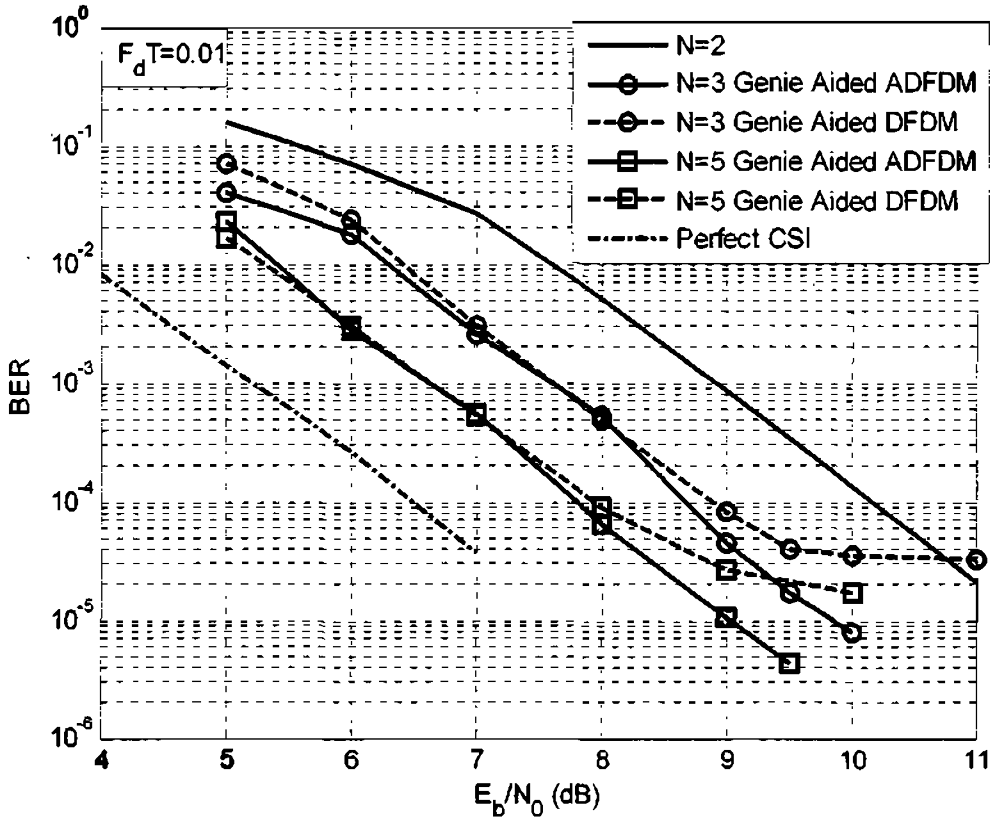

Fig. 5.1 Comparative genie aided BER curves for the adaptive DFDM and the DFDM.

In Fig. 5.1 only the genie aided cases are investigated to see if there is a possible improvement in the ideal feedback bound. It is seen that the genie aided case for the adaptive DFDM (ADFDM in the graphs) shows improvement for both observation windows $N = 3$ and $N = 5$, with a reduction in the error floor.

The performance of DFDM was achieved using the optimal statistical coefficients for the channel, which is assumed known at the receiver. However, the adaptive DFDM scheme adapts to the actual channel variations, therefore with the noise and fading variances being large in comparison to that of the DFDD, the statistically determined fixed coefficients do not achieve optimal performance for the DFDM. It was shown that the metrics are mathematically equivalent, therefore the only difference between the adaptive and fixed metric are the filter coefficients. Since the adaptive DFDM results are better than the DFDM, the adaptively determined coefficients are more optimal than the fixed coefficients determined statistically.

In Fig. 5.2, the performance of the adaptive DFDM with decision feedback symbols is shown. The plot displays a comparison between the adaptive DFDM and the DFDM. The results for adaptive DFDM are achieved after 5 iterations, while those for the DFDM after 4, just as in [23]. If the training iteration is excluded from the iteration count then the adaptive DFDM should exhibit similar performance gains after the same number of iterations as the DFDM. It is for this reason that the number of iterations used for the adaptive DFDM was 5.

As can be seen in Fig. 5.2, like the genie aided case the adaptive scheme performs better than the standard DFDM for both observation windows, $N = 3$ and $N = 5$. It was also noted that at the lower spectrum of SNRs, for observation windows $N > 2$, the performance diverges from the $N = 2$ case for the standard DFDM. The adaptive DFDM on the other hand does not, and remains below that of the $N = 2$ case for both observation windows, remaining more stable than the DFDM.

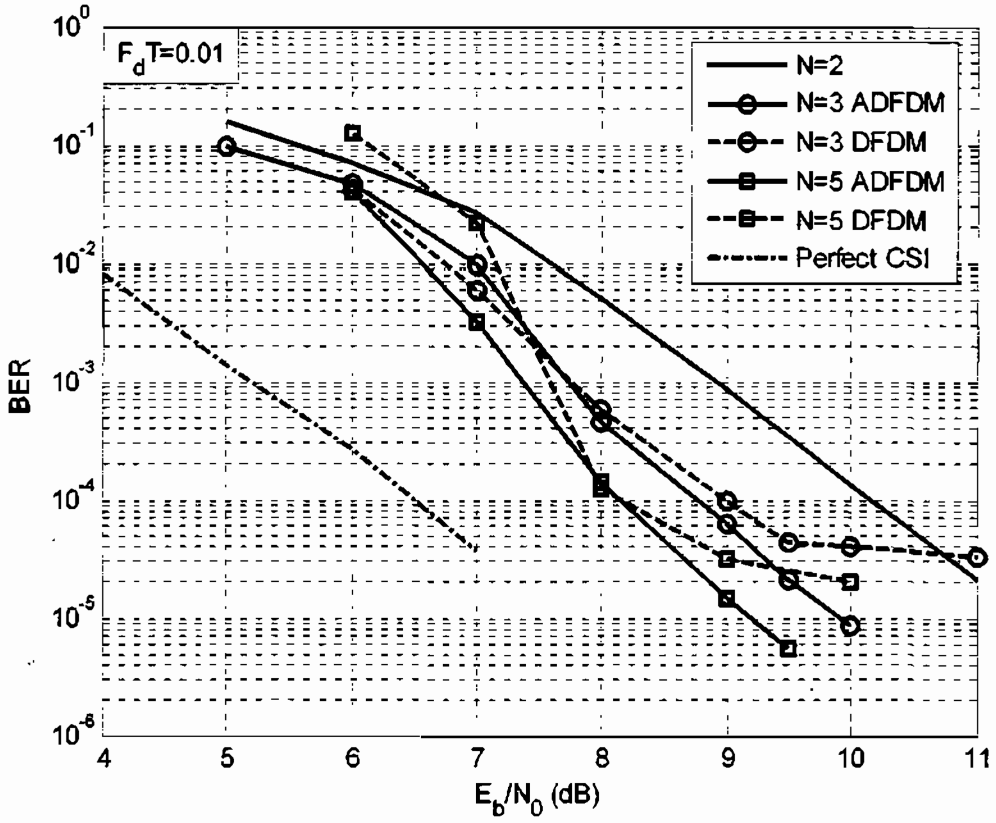

Fig. 5.2 Comparative BER curves for the adaptive DFDM and the DFDM with decision feedback after 5 and 4 iterations respectively.

The adaptively determined coefficients are yet again the only difference between the DFDM and the adaptive DFDM schemes, while the simulation parameters remained constant. The adaptive DFDM offers stability at the low SNRs, while the error floor is reduced and convergence is achieved for a larger range of SNRs.

In [23], it was seen that for larger observation windows the performance in the low SNR region is expected to be worse than smaller observation windows through the convergence analysis. In the case of the adaptive DFDM, the adaptive process seems to compensate for the larger noise levels, and allows the adaptive DFDM to maintain performance better than that of the standard BICM.

In Fig. 5.1 and Fig. 5.2, the adaptive DFDM was compared to its fixed coefficient counterpart, the DFDM. It was seen that the adaptive DFDM offers stability at low SNRs and a reduction in the error floor. Based on these results it is worth investigating the filter coefficients that are achieved after the training iteration.

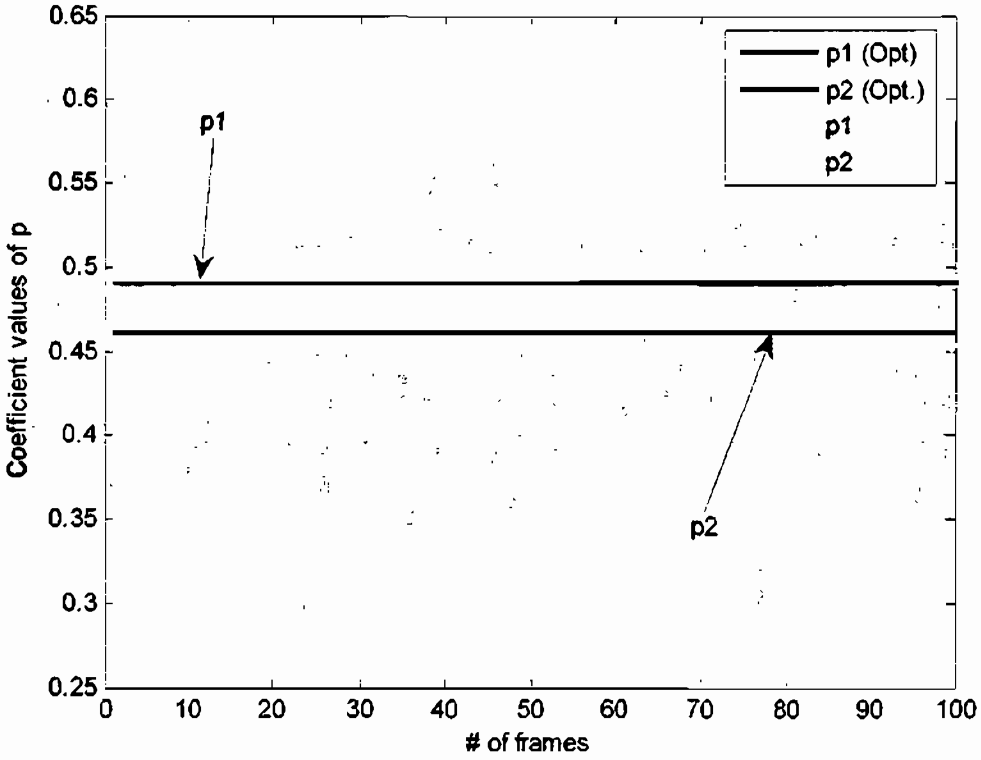

Fig. 5.3 The filter coefficients reached over 100 frames of training for N=3 with F_dT=0.01 at 10dB. The optimal Wiener Coefficients are also shown

In Fig. 5.3, the filter coefficients that are reached after the training iteration are compared to the optimal Wiener solution. It can be seen that the values do not always converge to the Wiener solution. The results were shown for an SNR of 10dB, where the fading dominates the performance. It was calculated that the mean of the trained coefficients after 1000 training iterations were 0.4674 and 0.4653 for $\hat{p}_1$ and $\hat{p}_2$ respectively. This is close to the optimal normalized solution of $p_1 = 0.5144$ and $p_2 = 0.4378$, however, the standard deviation about the mean value was calculated to be 0.04, which is relatively large. The performance improvement is therefore due to the fact that the statistics do not accurately depict the optimum performance per frame. The adaptive scheme is trained to the optimal solution for each frame, and hence better performance is achieved. It was seen in Fig. 5.3, that the coefficients deviated a great deal from the optimal solution in some cases.

These results reinforce the necessity for accurate statistics for even a coded decision feedback scheme. In the case of the adaptive DFDM a superior performance is achieved because in general the filter coefficients are more optimal for the frame instead of the statistically determined fixed coefficients.

In Fig. 5.4 the convergence of adaptive DFDM scheme is seen after subsequent iterations. It is seen that for the training iteration there is virtually no difference between the $N=3$ and $N=5$ curves. The majority of the gain occurs in the iteration after training and in smaller increments thereafter. As noted before, the iterations in general seem to converge in areas where the standard DFDM diverges for larger observation windows. The solid lines are for observation window $N=3$, while the dashed lines represent observation window $N=5$. The genie aided bounds are also shown to illustrate the achievable performance gains through normal decision feedback.

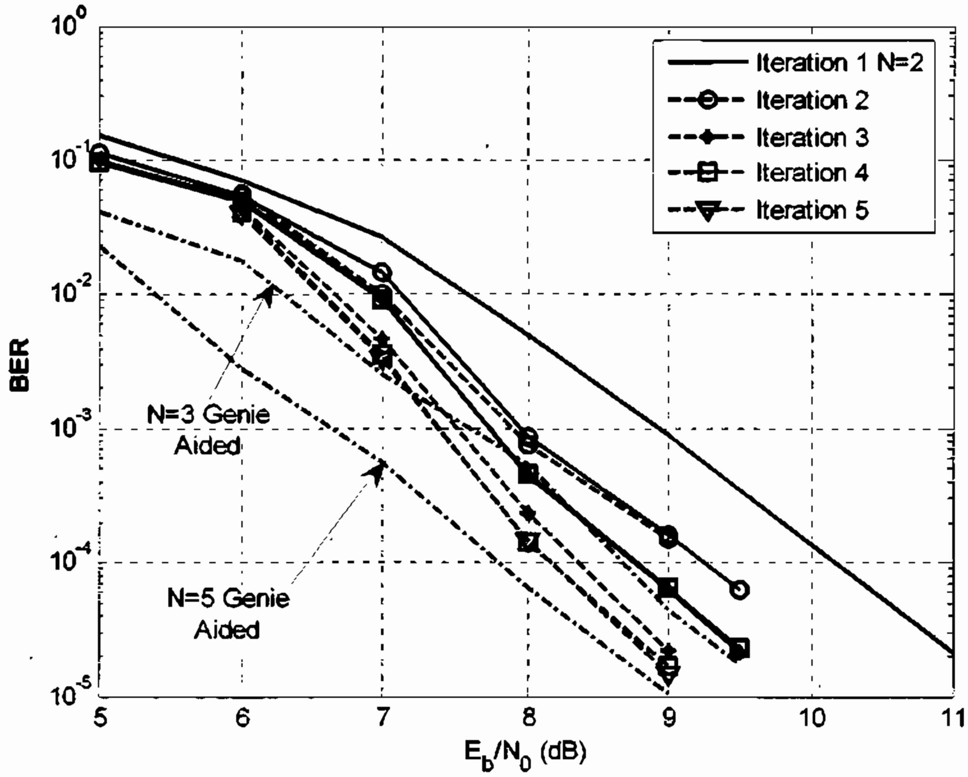

Fig. 5.4 Plot of BER convergence for adaptive DFDM, 4 iterations for N=3 (solid lines), while 5 iterations for N=5 (dashed lines).

In Fig. 5.5, the results for the adaptive DFDM are summarized, and it can be seen that the decision feedback curves converge to the genie aided case. It is seen that the adaptive DFDM with decision feedback performs better than the DFDM genie aided bound for both observation intervals at high SNRs due to the flooring of the DFDM. A similar occurrence of early flooring occurs in the DFDD with even marginally suboptimum coefficients as illustrated in chapter 3.

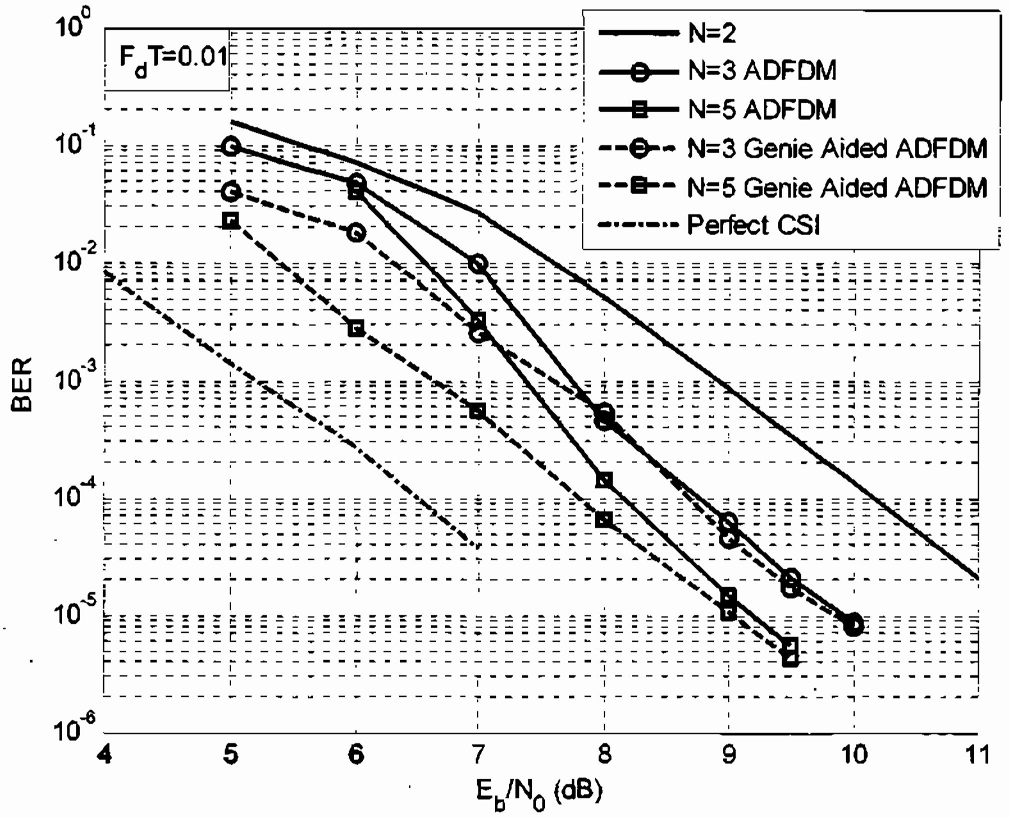

Fig. 5.5 BER of adaptive DFDM with genie aided performance bounds. Coherent detection with perfect CSI is also shown.

5.7 Conclusion

The new blind iterative DFDM scheme performs in a more robust manner than the standard DFDM. Not only did it remain convergent at the lower SNRs, but it also performed better at the high SNRs.

The adaptive DFDM has increased complexity due to the RLS algorithm, but this occurs only for one iteration, while the remaining iterations have the same complexity, since a fixed coefficient is used. This adaptive DFDM does not require any channel state information, nor does it require the channel statistics, like the DFDM. In addition the training does not require pilot symbols therefore the adaptation is blind.

The convergence and performance gains were confirmed through simulation. This new iterative structure of adaptive DFDM may be utilized in alternate iterative decision feedback detection structures.

The adaptive DFDM scheme maintained the advantages of the DFDM, while improving the overall performance, and making the performance robust at lower SNRs.

Chapter 6
Conclusion

6.1 Conclusion of Dissertation

The idea behind this dissertation was to investigate non-coherent detection through the use of multiple symbol detection (MSD) for flat fading channels. A novel and simple scheme employing decision feedback over a multiple symbol observation window, proved an effective strategy for the frequency flat Rayleigh environment, often encountered in mobile wireless channels. This system was the DFDD introduced in [21].

The core of the work presented in this dissertation revolves around this decision feedback principle. In chapter 2, relevant literature involving MSD was reviewed, and the DFDD principle stood out as a favourable MSD technique. The literature covered papers for the uncoded differential detection; the channel encoded differential detection schemes; as well as some multi-antennae differential schemes. The Linear filter theory for linear prediction filters were additionally summarized since these were employed in the DFDD, and the other DFDD schemes. The DFDD, the alternate linear prediction DFDD scheme of [22], and the associated adaptive DFDD scheme using the RLS algorithm were summarized and replicated through simulation in chapter 3. These performance results were later used for comparison purposes. The performance penalty of inaccurate channel statistics was also seen and noted. The equivalence of the DFDD ([21]), and the alternate DFDD ([22]), for Rayleigh channels was derived. To the authors knowledge this equivalence was not shown in any published literature reviewed.

An alternate adaptive scheme is proposed in chapter 4 for uncoded DFDD. In this new adaptive algorithm, the numerically stable, pipelined QR decomposition was utilized to offer a stable and efficient algorithm for differential detection. This system is referred to as QR-DFDD. However, memory requirements were inhibitive, therefore an alternate simpler QR based algorithm using sliding windows was introduced. The sliding window DFDD algorithm (QRSW-DFDD), was less memory intensive, and still maintained the highly pipelined architecture associated with the QR decomposition.

The performance of QR-DFDD and QRSW-DFDD was suboptimal in normal flat fading conditions, but remained comparable to the DFDD and the adaptive DFDD algorithms. The QRSW-DFDD proved superior to the adaptive DFDD in [22], referred to as RLS-DFDD, in the time selective fading channel. The performance was near the optimal DFDD performance in these conditions. In fast fading conditions the QR based algorithms are advantageous. The channel varies rapidly in time, and the channel statistics are harder to determine in this case. The adaptive algorithms do not require training or any pilot symbols, hence the adaptation is blind.

In summary the QR based DFDD algorithms offered all round performance that was comparable to the DFDD, for time selective or time non-selective fading. The QR based algorithms were simpler to compute, offered highly pipelined implementations, and were suitable to VLSI. The adaptive process is blind, therefore the performance does not need any channel information, statistics or pilot symbols.

An iterative coded system utilizing the DFDD principle was introduced in [23], called iterative decision feedback differential demodulation (DFDM). The DFDM offered large performance gains, while using metrics that were computationally simple for frequency flat fading channels. In chapter 5, an alternate approach to DFDM was introduced; it utilized the RLS algorithm in an adaptive scheme to maximize the gains of DFDM, over conventional bit interleaved coded non-coherent detection.

An alternative iterative decoding procedure was proposed, which required the application of the RLS algorithm for a single training iteration, minimizing the additional complexity, on the adaptive DFDM. The performance was expected to be similar to that of the DFDM, but the adaptive DFDM scheme proposed offered improved performance at higher SNRs, while maintaining stability at low SNRs. An additional advantage of this scheme was that the channel statistics were no longer required. The adaptive scheme was therefore totally ignorant of the channel. It also had the same properties of the RLS-DFDD, where no pilot symbols are required, and hence the adaptive process was blind.

The aim of this dissertation was to investigate MSD. The pertinent literature in this area has been reviewed. The decision feedback schemes that form the foundation of this dissertation were summarized and replicated through simulation. Two new adaptive approaches to non-coherent detection using MSD and decision feedback were introduced. The performances of these systems were compared to relevant literature. The advantages of these new schemes were highlighted.

6.2 Future Work

Theoretical performance analysis of the iterative DFDM would be useful. However, the combination of interleaving, convolutional coding, decision feedback and non-coherent detection offers many difficulties.

The application of these adaptive algorithms has been limited to the single user, and the single antenna cases. These adaptive algorithms may offer improved performance in multi-antenna systems. In deep fades the concept of cooperative diversity is another diversity method that may be employed.

The application of MSD has been restricted to the AWGN channel, the Rayleigh fading channel or the Ricean channel. This is due to the complexity of the analysis using other models. Alternate channel models may be used and investigated e.g. Nakagami m-distributions. The application of multiple symbol decision feedback for channel equalization may also be a subject of interest.

An alternate MSD technique for differential detection, using the Karhunen-Loève expansion was described in [28]. It may be possible to exploit this expansion to employ the decision feedback principle.

The reduced complexity algorithm for multiple symbol differential detection was introduced in [26]. This may be applied to the MSD techniques available for fading channels. The complexity should be comparable to that of [29] in the uncoded case. It may also be introduced to reduce the trellis complexities in [15, 34].

In conclusion, the results are very encouraging for the application of adaptive algorithms in MSD non-coherent detection. These other research topics may prove fruitful areas for future work.

Appendices

A: Rayleigh Fading Generation Method

Using the following equations an accurate Rayleigh Fading generator is developed. M is the number of sinusoids, F_d the maximum Doppler frequency and T the symbol period. This is based on the Jakes Model outlined in [47].

Assuming $N = 4M + 2$, and $w_d = 2\pi F_d T$, the fading gain is

$$h[k] = u_r[k] + j u_I[k], \tag{A.1}$$

where

$$u_r[k] \square \frac{2}{\sqrt{N}} \sum_{n=1}^{M} a_n \cos(w_n[k]) + \sqrt{2}\cos(\pi/4)\cos(w_d), \tag{A.2}$$

and

$$u_I[k] \square \frac{2}{\sqrt{N}} \sum_{n=1}^{M} b_n \cos(w_n[k]) + \sqrt{2}\sin(\pi/4)\cos(w_d). \tag{A.3}$$

The coefficients are defined as

$$\left.\begin{array}{l} a_n = 2\cos\beta_n \\ b_n = 2\sin\beta_n \end{array}\right\} n = 1,2,\ldots,M, \tag{A.4}$$

and

$$\beta_n = \frac{\pi n}{M}, n = 1,2,\ldots,M, \tag{A.5}$$

with

$$w_n = w_d \cos(\frac{2\pi n}{N}). \tag{A.6}$$

Bibliography

[1] J. G. Proakis, *Digital Communications*, 4th ed. New York: Mcgraw-Hill, 2001.

[2] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM Journal on Applied Mathematics*, vol. 8, pp. 300-304, 1960.

[3] P. Elias, "Coding for Noisy Channels," *IRE Conv. Record*, vol. 4, pp. 37-47, 1955.

[4] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Comm.*, vol. 13, pp. 260-269, Apr. 1967.

[5] G. D. Forney, "Concatenated codes," *Cambridge, MIT Press*, 1966.

[6] C Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," in Proceedings, IEEE ICC, Geneva, Switzerland, May 1993.

[7] S. Benedetto and G. Montorsi, "Serial Concatenation of Block and Convolutional Codes," *IEE Electronics Letters*, vol. 32, No. 16, pp. 887-888, May 1996.

[8] S. Benedetto and G. Montorsi, "Iterative Decoding of Serially Concatenated Convolutional Codes," *IEE Electronics Letters*, vol. 32, No. 13, pp. 1186-1188, Jun. 1996.

[9] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative Correction of Intersymbol Interference: Turbo Equalization," *European Trans. on Telecomm.*, vol. 6, pp. 507-511, Sept.-Oct. 1995.

[10] M. C. Reed, C. B. Schlegel, P. Alexander, and J. Asenstorfer, "Iterative Multi-user Detection for DS-CDMA with FEC," in Proceedings, International Symposium of Turbo Codes, France, Sept. 1997.

[11] P. Jung, M. Nasshan, and J. Blanz, "Application of Turbo-Codes to a CDMA Mobile Radio System using Joint Detection and Antenna Diversity," in Proceedings, IEEE 44th Vehicular Technology Conference, Stockholm, June 1994.

[12] K. R. Narayanan and G. L. Stüber, "A Novel ARQ Technique using the Turbo Coding Principle," *IEEE Commun. Lett.*, vol. 1, pp. 49-51, Mar. 1997.

[13] J. Garcia-Frias and J. D. Villasenor, "Joint Source Channel Decoding of Turbo Codes," in Proceedings, International Symposium on Turbo Codes, France, Sept. 1997.

[14] I. D. Marsland, "Iterative Noncoherent Detection of Differentially Encoded M-PSK," in *Dept. of Electrical and Computer Engineering*, vol. Doctor of Philosophy: University of British Columbia, 1999.

[15] P. Hoeher and J. Lodge, ""Turbo DPSK"; Iterative differential PSK demodulation and channel decoding," *IEEE Trans. on Comm.*, vol. 47, pp. 837-843, Jun. 1999.

[16] H. Taub and D. L. Schilling, *Principles of Communication Systems*, 2nd ed. New Delhi: Tata McGraw-Hill, 2000 (15th Reprint).

[17] H. Xu, "Chapter 1: Modeling of Wireless Channels," in *Class Notes*. Durban, 2004.

[18] S. Haykin, *Communication Systems*, 3 ed: John Wiley & Sons, Inc., 1994.

[19] B. M. Hochwald, T. L. Marzetta, T. L. Richardson, W. Sweldens, and R. Urbanke, "Systematic Design of Unitary Space-Time Constellations," *IEEE Trans. on Inform. Theory.*

[20] D. Divsalar and M. K. Simon, "Multiple Symbol Differential Detection of MPSK," *IEEE Trans. on Comm.*, vol. 38, No. 3, pp. 300-308, Mar. 1990.

[21] R. Schober, W. H. Gerstacker, and J. H. Huber, "Decision Feedback Differential Detection of MDPSK for Flat Fading Channels," *IEEE Trans. on Comm.*, vol. 47, pp. 837-843, Jul. 1999.

[22] R. Schober and W. H. Gerstacker, "Decision-Feedback Differential Detection Based on Linear Prediction for MDPSK Signals Transmitted over Ricean Fading Channels," *IEEE J. Select. Areas Commun.*, vol. 18, No. 3, pp. 391-402, Mar. 2000.

[23] L. H.-J. Lampe and R. Schober, "Iterative Decision-Feedback Differential Demodulation of Bit-Interleaved Coded MDPSK for Flat Rayleigh Fading Channels," *IEEE Trans. on Comm.*, vol. 49, No. 7, pp. 1176-1184, Jul. 2001.

[24] H. Leib and S. Pasupathy, "The Phase of a Vector Perturbed by Gaussian Noise and Differentially Coherent Receivers," *IEEE Trans. on Inform. Theory*, vol. 34, No. 6, pp. 1491-1501, Nov. 1988.

[25] F. Edbauer, "Bit Error Probability of Binary and Quarternary DPSK signals with Multiple Differential Detection," *IEEE Trans. on Comm.*, vol. 40, No. 3, pp. 457-460, Mar. 1992.

[26] B. Li, "A New Reduced Complexity Algorithm for Multiple Symbol Differential Detection," *IEEE Commun. Lett.*, vol. 7, No. 6, pp. 269-271, Jun. 2003.

[27] P. Ho and D. Fung, "Error Performance of Multiple Symbol Differential Detection of PSK Signal Transmitted Correlated Rayleigh Fading channels," *IEEE Trans. on Comm.*, vol. 42, No. 10 pp. 1566-1569, Oct. 1992.

[28] M. Visintin, "Differential PSK Block Demodulation over a Flat Correlated Rayleigh Fading Channel," *IEEE Trans. on Comm.*, vol. 45, No. 1, pp. 9-11, Jan. 1997.

[29] L. H.-J. Lampe, R. Schober, V. Pauli, and C. Windpassinger, "Multiple Symbol Differential Sphere Decoding," *IEEE Trans. on Comm.*, vol. 53, No. 12, pp. 1981-1985, Dec. 2005.

[30] Y. Liu and X. Wang, "Multiple Symbol Decision Feedback Space-Time Differential Decoding in Fading Channels," *EURASIP Journal on Applied Signal Processing*, vol. 3, pp. 297-304, 2002.

[31] G. Caire, G. Tarrico, and E. Biglieri, "Bit-interleaved Coded Modulation," *IEEE Trans. on Comm.*, vol. 44, pp. 927-946, May 1998.

[32] X. Li and J. A. Ritcey, "Trellis-coded modulation with bit interleaving and iterative decoding," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 715-724, Apr. 1999.

[33] L. H.-J. Lampe and R. Schober, "Decision-feedback differential demodulation of bit-interleaved coded MDPSK," *Electronics Letters*, vol. 35, No. 25, pp. 2170-2171, Dec. 1999.

[34] G. Colavolpe and R. Raheli, "Noncoherent Sequence Detection," *IEEE Trans. on Comm.*, vol. 47, pp. 1376-1385, Sept. 1999.

[35] G. Colavolpe and R. Raheli, "Theoretical Analysis and Performance Limits of Noncoherent Sequence Detection of Coded PSK," *IEEE Trans. on Inform. Theory*, vol. 46, No. 4, pp. 1483-1494, Jul. 2000.

[36] L. H.-J. Lampe and R. Schober, "Improvement and Analysis of Iterative Decision Feedback Differential Demodulation," *IEEE Trans. on Comm.*, vol. 53, No. 8, pp. 1263-1268, Aug. 2005.

[37] G. Colavolpe, G. Ferrari, and R. Raheli, "Noncoherent iterative (turbo) decoding," *IEEE Trans. on Comm.*, vol. 48, pp. 1488-1498, Sept. 2000.

[38] L. H.-J. Lampe, V. Pauli, and R. Schober, ""Turbo DPSK" Using Sphere Decoding without Channel State Information," in Proceedings, *International Symposium on Information Theory and its Applications, ISITA 2004*. Parma, Italy, Oct. 2004.

[39] L. H.-J. Lampe and R. Schober, "Bit-Interleaved Coded Differential Space Time Modulation," *IEEE Trans. on Comm.*, vol. 50, No. 9, pp. 1429-1439, Sept. 2002.

[40] L. H.-J. Lampe, R. Schober, and R. Fischer, "Coded Differential Space-Time Modulation for Flat Fading Channels," *IEEE Trans. on Wireless Comms.*, vol. 2, No. 6, pp. 582-590, May 2003.

[41] S. Haykin, *Adaptive Filter Theory*. New Jersey: Prentice-Hall, 1986.

[42] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*: Prentice-Hall, 2000.

[43] R. J. Young and J. H. Lodge, "Detection of CPM Signals in Fast Rayleigh Flat Fading Using Adaptive Channel Estimation," *IEEE Trans. on. Veh. Tech.*, vol. 44, No 2, pp. 338-347, May 1995.

[44] Y. Liu and S. D. Blostein, "Identification of Frequency Non-selective Fading Channels using Decision Feedback and Adaptive Linear Prediction," *IEEE Trans. on Comm.*, vol. 43, pp. 1484-1492, Feb./Mar./Apr. 1995.

[45] R. Chen, X. Wang, and J. S. Liu, "Adaptive Joint Detection and Decoding in Flat Fading Channels via Mixture Kalman Filtering," *IEEE Trans. on Inform. Theory*, vol. 46, pp. 2079-2094, Sept. 2000.

[46] X. Wang and H. V. Poor, *Wireless Communication Systems - Advanced Techniques for Signal Reception*. Upper Saddle River, NJ.: Prentice Hall, 2004.

[47] W. C. Jakes, *Microwave Mobile Communications (IEEE Classic Reissue)*. New York: IEEE Press, 1994.

[48] M. Peleg and S. Shamai, "Iterative decoding of coded and interleaved non-coherent multiple symbol detected DPSK," *Electronics Letters*, vol. 33, No. 12, pp. 1018-1020, Jun. 1997.

[49] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, pp. 169-171, Nov. 1999.

[50] D. Makrakis, T. Mathiopoulos, and D. P. Bouras, "Optimal Decoding of Coded PSK and QAM Signals in Correlated Fast Fading Channels and AWGN: A Combined Envelope,

Multiple Differential and Coherent Detection Approach," *IEEE Trans. on Comm.*, vol. 42, No. 1, pp. 63-75, Jan. 1994.

[51] M. Mahon, L. Sibul, and H. Valenzuela, "A Sliding Window Update for the Basis Matrix of the QR Decomposition," *IEEE Trans. Signal Processing,* vol. 41, pp. 1951-1953, May 1993.

[52] B. Baykal and A. G. Constantinides, "Sliding Window Adaptive Fast QR and QR-Lattice Algorithms," *IEEE Trans. Signal Processing,* vol. 46, No. 11, pp. 2877-2887, Nov. 1998.