SIC	Successive/Serial Interference Cancellation
SIR	Signal-to-Interference Ratio
SOVA	Soft Output Viterbi Algorithm
SNR	Signal-to-Noise Ratio, SNR = E_b/N_0
SR	Shift Register
TDMA	Time Division Multiple Access
TH-CDMA	Time Hopping CDMA
W-CDMA	Wideband CDMA

BLIND ITERATIVE MULTIUSER DETECTION FOR ERROR CODED CDMA SYSTEMS

Brett van Niekerk

Submitted in fulfilment of the requirements for the Degree of Master of Science in Engineering in the Centre for Radio Access Technologies, School of Electrical, Electronic and Computer Engineering at the University of KwaZulu-Natal, Durban, under the Graduate Assistantship Programme.

August 2005.

ABSTRACT

Mobile communications have developed since the radio communications that were in use 50 years ago. With the advent of GSM, mobile communications was brought to the average citizen. More recently, CDMA technology has provided the user with higher data rates and more reliable service, and it is apparent that it is the future of wireless communication. With the introduction of 3G technology in South Africa, it is becoming clear that it is the solution to the country's wireless communication requirements. The 3G and next-generation technologies could provide reliable communications to areas where it has proven difficult to operate and maintain communications effectively, such as rural locations. It is therefore important that these technologies continue to be researched in order to enhance their capabilities to provide a solution to the wireless needs of the local and global community.

Whilst CDMA is proving to be a reliable communications technology, it is still susceptible to the effects of the near-far problem and multiple-access interference. A number of multiuser detectors have been proposed in literature that attempt to mitigate the effects of multiple-access interference. A notable detector is the blind MOE detector, which requires only the desired user's spreading sequence, and it exhibits performance approximating that of other linear multiuser detectors. Another promising class of multiuser detector operate using an iterative principle and have a joint multiuser detection and error-correcting coding scheme. The aim of this research is to develop a blind iterative detector with FEC coding as a potential solution to the need for a detector that can mitigate the effects of interfering users operating on the channel. The proposed detector has the benefits of both the blind and iterative schemes: it only requires the knowledge of the desired user's signature, and it has integrated error-correcting abilities. The simulation results presented in this dissertation show that the proposed detector exhibits superior performance over the blind MOE detector for various channel conditions.

An overview of spread-spectrum technologies is presented, and the operation of DS-CDMA is described in more detail. A history and overview of existing CDMA standards is also given. The need for multiuser detection is explained, and a description and comparison of various detection methods that have appeared in literature is given. An introduction to error coding is given, with convolutional codes, the turbo coding concept and methods of iterative detection are described in more detail and compared, as iterative decoding is fundamental to the operation of an iterative CDMA detector. An overview of iterative multiuser detection is given, and selected iterative methods are described in more detail.

A blind iterative detector is proposed and analysed. Simulation results for the proposed detector, and a comparison to the blind MOE detector is presented, showing performance characteristics and the effects of various channel parameters on performance. From these results it can be seen that the proposed detector exhibits a superior performance compared to that of the blind MOE detector for various channel conditions. The dissertation is concluded, and possible future directions of research are given.

PREFACE

The research work presented in this thesis was performed by Mr. Brett van Niekerk, under the supervision of Prof. Stanley Mneney at the Centre for Radio Access Technologies, School of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal, Durban. This work was partially sponsored by Alcatel Altech Telecoms and Telkom SA Ltd as part of the Centre of Excellence programme.

The entire dissertation, unless otherwise indicated, is the original work of the student and has not been submitted in part, or in whole, to any other institution for degree purposes.

Signed:_____

Name: _____

Date:_____

ACKNOWLEDGEMENTS

I would like to express my gratitude towards Prof. S. H. Mneney for his guidance and assistance throughout the course of this project. He gave me the freedom to approach the project with my own methods, and provided technical inputs and encouragement when needed. His willingness to set aside time to assist me has been most appreciated. I also thank him for presenting our work at IEEE Africon 2004.

Thanks are owed to Telkom SA Ltd and Alcatel Altech Telecoms for their much appreciated financial support through the Centre for Radio Access Technologies, part of the Centre of Excellence program, and the Graduate Assistantship Program of the University of KwaZulu-Natal.

I would also like to thank the support staff and my postgraduate colleagues for their help and friendship, creating a most enjoyable working environment. Finally, I would like to thank my family and friends who supported me through this project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

3G	Third Generation
AM	Amplitude Modulated
AMPS	Advanced Mobile Phone Service
AWGN	Additive White Gaussian Noise
BCH	Bose, Chaudhuri, and Hocquenghem
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
DS-CDMA	Direct Sequence CDMA
EDGE	Enhanced Data rates for Global/GSM Evolution
FDMA	Frequency Division Multiple Access
FH-CDMA	Frequency Hopping CDMA
FM	Frequency Modulated
GSM	Global System for Mobile communications
GPRS	General Packet Radio Service
HSCSD	High Speed Circuit Switched Data
IC	Interference Cancellation
IEEE	Institute for Electrical and Electronic Engineers
IIR	Infinite Impulse Response
LAPP	Log *a posteriori* probability
LLR	Log Likelihood Ratio
LMS	Least Mean Squares
MAI	Multiple Access Interference
MAP	Maximum *a posteriori*
ML	Maximal Length / Maximum Likelihood
MMSE	Minimum Mean Squared
MOE	Minimum Output Energy
MUD	Multiuser Detector
PIC	Parallel Interference Cancellation
PN	Pseudo Noise
RLS	Recursive Least Squares
RS	Reed-Solomon
RSC	Recursive Systematic Convolutional

SIC	Successive/Serial Interference Cancellation
SIR	Signal-to-Interference Ratio
SOVA	Soft Output Viterbi Algorithm
SNR	Signal-to-Noise Ratio, SNR = E_b/N_0
SR	Shift Register
TDMA	Time Division Multiple Access
TH-CDMA	Time Hopping CDMA
W-CDMA	Wideband CDMA

CHAPTER 1 INTRODUCTION

The world of mobile communications has advanced over the last 50 years. Initially commercial mobile technologies were analogue and only started making a large appearance in the 1980s, then a series of second-generation systems using digital transmission were commercialised. These new systems offered higher spectrum efficiency, better data services and more advanced roaming. The most popular of these technologies is GSM (Global System for Mobile Communications), which is still in widespread service today throughout Africa, Europe and the Middle East. The GSM standard was first conceptualised in 1982, the goal being to develop a Pan-European cellular mobile network [5]. The official launch of GSM into the commercial world was in 1992. Since then, an updated second phase of GSM was launched in 1996, with higher data rates that could support technologies such as HSCSD (high speed circuit switched data), GPRS (general packet radio service), and EDGE (enhanced data rates for global/GSM evolution). These newer technologies were called generation 2.5. Recently, the third generation of mobile communications, based on the CDMA (spread spectrum) technology was launched. The 3G technologies offer even higher data rates, more flexibility and different quality of service classes [5].

The origins of spread spectrum technologies lie in the military field and navigation systems, and techniques developed to mitigate the effects of intentional jamming have proved suitable for cellular applications, where communications may be through dispersive channels [5]. John Pierce wrote a technical memorandum in 1949, where he described a multiplexing system that can be classified as a time hopping spread spectrum multiple access system, in which a common medium carries coded signals that need not be synchronous, [5]. In 1949 Claude Shannon and Robert Pierce introduced the basic ideas of CDMA by describing the interference averaging effect and the graceful degradation of CDMA. A direct sequence spread spectrum system was proposed in 1950 by De Rosa-Rogoff, where the noise multiplexing idea and processing gain equation was introduced [5]. Price and Green filed a patent for the RAKE, an anti-multipath receiver, in 1956. Magnuski mentioned the near-far problem in 1961. Cooper and Nettleton proposed the cellular application of spread spectrum systems in 1978, and in July 1993 spread spectrum communications was commercialised in the form of the narrowband CDMA IS-95 standard, which began commercial operation in 1996 [5]. During the 1990s, wideband CDMA techniques were studied intensively, resulting in a number of schemes, most notably CDMA 2000 and WCDMA. Recently, third generation wireless systems were introduced using wideband CDMA.

Extensive research into multiuser detection (MUD) began when Verdu formulated the maximum likelihood sequence estimator (MLSE), which is an optimum detector for additive white Gaussian noise (AWGN) channels [5]. A number of sub-optimal receivers have since appeared in literature, using various detection techniques to mitigate multiple access interference (MAI) and the effects of multipaths and the near-far problem. The evolution of CDMA communications is summarised in Table 1.1 [5]:

Table 1.1: CDMA Era

Pioneer Era	
1949	John Pierce: time hopping spread spectrum Claude Shannon & Robert Pierce: basic ideas of CDMA
1950	De Rosa-Rogoff: direct sequence spread spectrum
1956	Price and Green: RAKE receiver
1961	Magnuski: the near-far problem
1970s	Developments in the military field and navigation systems
Narrowband CDMA Era	
1978	Cooper and Nettleton: concept of the cellular application of spread spectrum
1980s	Investigation of narrowband CDMA techniques for cellular application
1986	Verdu: Optimal multiuser detector
1993	IS-95 standard
Wideband CDMA Era	
1995-	Wideband CDMA in Europe, Japan and United States
2000s	Commercialisation of wideband CDMA systems

1.1 Motivation

With the introduction of 3G cellular systems in South Africa, it is apparent that such systems provide superior data transfer rates compared to the established fixed-line telephone and GSM cellular systems. The characteristics of CDMA systems will also provide more stable connections for both voice and data. Whilst 3G technology has not been fully established in South Africa, it has the potential to provide the solution to a more reliable wireless communications infrastructure in both rural and urban areas. For this reason, it is imperative that 3G communications continue to be researched and enhanced, to evolve next-generation systems that can provide the solution to local and global wireless communication requirements.

Whilst DS-CDMA is becoming an ever popular multiple access technology for wireless communications, it is still not entirely immune from the effects of interference, and is particularly susceptible to multiple-access interference and the near-far effect. Multiuser detection was developed in an attempt to mitigate these effects, and a number of multiuser

detection schemes have been proposed in literature, each with its advantages and disadvantages. Linear detectors, such as the decorrelating and MMSE detectors, require information about the channel, the desired user and interfering users to estimate the interference due to cross-correlation and the channel, and thus can reduce the effect on the desired users signal. As will be seen in Chapter 3, blind detectors require no information about the interfering users or the channel, yet they perform comparably to other linear multiuser detectors. A common blind detector uses the minimum output energy (MOE) criterion, where the output variance is minimised with respect to components that are orthogonal to the desired user's spreading waveform.

Another class of multiuser detector is the iterative detector. They have the advantage of having an integrated multiuser detection/error coding architecture, and exhibit superior performance over their non-iterative counterparts. Iterative multiuser detectors operate by passing information from soft-input soft-output (SISO) CDMA decoders to SISO error decoders, which feeds the SISO CDMA decoder updated information for the next iteration. This principle of operation is similar to that of the iterative decoding methods used in turbo codes.

The goal of this research is to develop a detector that has improved resistance to the effects of interference, multipaths and the near-far problem. The proposed solution to these problems is to combine the blind MOE and iterative techniques, resulting in a blind iterative detector that will require the least amount of information for multiuser detection, and still provide error-correcting capabilities to further improve performance. Whilst most iterative systems utilise the MAP algorithm to perform iterative decoding, the proposed detector uses the soft-output Viterbi algorithm to reduce computational complexity.

1.2 Organisation of the Thesis

The report is divided into three main sections: Chapters 2 and 3 deal with spread-spectrum communications, CDMA and multiuser detection. Chapter 4 deals with error coding. Chapters 5 and 6 introduce the proposed iterative detector and present simulation results, respectively. A more detailed summary of the chapters is given below.

Chapter 2 describes the three multiple access communication technologies, frequency-division, time-division, and code-division multiple access schemes. An introduction to the principles of operation for DS-CDMA is given. An overview of existing CDMA technologies and their operating parameters is given.

Chapter 3 explains the need for multiuser detection, and describes various multiuser detection methods that have appeared in literature, but the primary focus of this chapter is on the blind

MOE scheme. The performance of the main linear detectors, being the conventional, decorrelating, MMSE and blind detectors are analysed and compared using both the bit error rate and asymptotic efficiency criterion. An overview of interference cancellation detectors is given as some iterative detectors described in Chapter 5 utilise the interference cancellation method.

Chapter 4 describes and compares various error coding schemes, with its focus being on convolutional codes and turbo codes, which will be required for the iterative detector. The two main iterative decoding techniques, the maximum *a posteriori* and soft-output Viterbi algorithms, are explained, as these techniques are fundamental to the operation of iterative CDMA detectors.

An overview of iterative multiuser detection is given in Chapter 5, with a few specific detectors being described in more detail. The proposed scheme is introduced, with the variations from previous blind iterative decoders being described. A theoretical analysis of the proposed detector is then given.

Chapter 6 presents simulation results for the proposed detector, with comparisons to the blind MOE detector, for general performance and to illustrate the effect various channel parameters have on performance. The results show an improved performance due to the iterative structure for the different operating conditions. The dissertation concludes in Chapter 7, where the key points are summarised and possible future research directions are given.

CHAPTER 2 DS-CDMA

This chapter presents an overview of multi-access communications, including TDMA and FDMA, and spread-spectrum systems, including TH-CDMA and FH-CDMA. The focus of the chapter is on DS-CDMA systems, which will be used as the transmission system for the proposed blind iterative detector. The general performance of DS-CDMA systems is simulated. An overview of PN sequences used for spreading codes and their characteristics is given. Three existing CDMA standards are described.

2.1 Multi-Access Communications

Multi-access communications is when multiple transmitters utilize a single channel, i.e. a channel is shared amongst numerous transmitters, and usually refers to situations where the sources are not collocated or operate autonomously. The sources are referred to as users [1]. There are many examples of multi-access communication systems: mobile telephones communicating with a base station, local-area networks, packet-radio networks, and satellite communications. Generally multi-access communications is multipoint-to-point.

Two common multiple-access methods are frequency-division multiple-access (FDMA) and time-division multiple-access (TDMA). In FDMA the channel bandwidth is divided up amongst the users, and in TDMA each user gets a timeslot. Each method has its advantages and disadvantages. In FDMA, the bandwidth each user utilizes is limited, but there is no need for time-synchronism. In TDMA, each user can use the entire bandwidth, but only transmits in "bursts", and time-synchronism is required, so all users must have access to a common clock. An important feature of both these methods are that the users are essentially operating in separate, non-interfering channels; hence the transmitted signals are mutually orthogonal.

The above two schemes ensure that there is no overlap in the time or frequency slots. If this were violated, the receivers would be unable to decode the signals that have overlapped. However, signals may overlap in both time and frequency: the key here is that their cross-correlation is zero:

$$\langle s_i, s_j \rangle = \int_0^T s_i(t) s_j(t) dt = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases} \tag{2.1}$$

This scheme is code-division multiple-access. The principle is that each user has a unique signature waveform that is orthogonal to the other users. Each user's transmitter modulates its own unique signature waveform as if it is a single-user communication system. CDMA communications may be synchronous or asynchronous, whereas TDMA systems have to be

synchronous. Figure 2.1 compares the bandwidth usage of TDMA, FDMA and CDMA systems for multiple users. Examples of wireless CDMA systems where the different users are not synchronous are: IS-95, CDMA2000, W-CDMA, BlueTooth (using frequency-hopping CDMA), and IEEE 802.11 [2], however the forward link of IS-95 and CDMA2000 are synchronous.

As the entire bandwidth is used for each signal, the bandwidth of the transmitted signal is greater than the bandwidth of the original message. For this reason this type of communication is known as spread spectrum, and will be discussed in Section 2.2 below.

2.2 Spread Spectrum Communications

As mentioned above, spread spectrum signals are distributed over the entire available bandwidth. The interference from the resulting waveform is barely noticeable, hence a receiver tuned to a specific AM or FM broadcast would not notice the existence of the overlapping spread spectrum signal, likewise the spread spectrum receiver would not notice the existence of the broadcast: the signals are said to be transparent [3].

There are three types of CDMA systems: frequency-hopping (FH-CDMA), time-hopping (TH-CDMA), and direct-sequence (DS-CDMA). For FH-CDMA N channels would be available for the system to hop over with a sequence determined by the spreading code. If the information channel has a bandwidth B_I, then the bandwidth of the FH signal is $B_{FH}=NB_I$. The processing gain of the system is [2]:

$$PG_{FH}=10\log N \text{ dB}. \tag{2.2}$$

There are two basic types of hopping sequences: fast hopping, which makes two or more hops per symbol, and slow hopping, where there are two or more symbols transmitted per hop.

TH-CDMA transmits data with a rate R, which originally required a time interval T, over a longer time interval T_s. The data is sent in bursts determined by the chip sequence. The time between bursts, t_n, can be varied, and the data rate of the spread signal, R_s, will always be lower than the initial data rate. For N bursts, occurring in time T:

$$R_s = \left(\frac{T_s}{T}\right)R = \left(1 - \frac{\sum_{n=1}^{N} t_n}{T}\right)R. \tag{2.3}$$

DS-CDMA systems represent each symbol by a series of chips (the spreading sequence). For a bit rate R, bandwidth B, and N chips, the chip rate is $R_c = RN$, and the DS bandwidth is $B_{DS} = BN$. The processing gain of a DS system is:

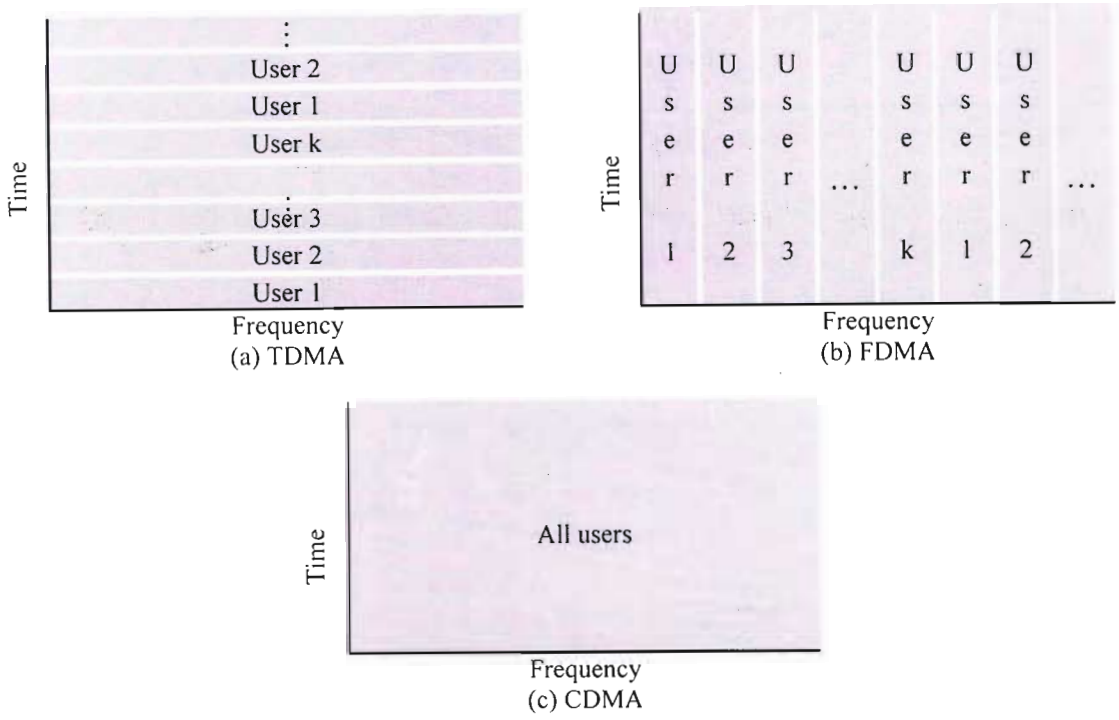


Figure 2.1: Comparison of bandwidth usage for TDMA, FDMA and CDMA systems

$$\mathrm{PG}_{\mathrm{DS}} = 10\log\frac{B_{\mathrm{DS}}}{B}\mathrm{dB}\,. \tag{2.4a}$$

For BPSK modulation $B_{DS}/B = (2/T_c)/(1/T_b) = 2T_b/T_c$, so the processing gain can be determined by:

$$\mathrm{PG}_{\mathrm{DS}} = \frac{2\mathrm{T_b}}{\mathrm{T_c}}, \tag{2.4b}$$

where T_b and T_c are the time intervals for a bit and a chip respectively. DS-CDMA systems will be discussed in greater detail in Section 2.3 below.

The spectral efficiency, E_{eff}, describes how efficiently the message is fitted into the transmission band [4]:

$$\begin{aligned} E_{eff} &= R/B_{DS}, \mathrm{G_p} = 2T_b/T_c \text{ (for BPSK)} \\ B_{DS} &\approx \frac{B_{DS,2-lev}}{k} = \frac{2/T_c}{\log_2 M} = \frac{\mathrm{G_p}}{T_b \log_2 M} \\ \Rightarrow E_{eff} &= \frac{\log_2 M}{\mathrm{G_p}} = \frac{k}{\mathrm{G_p}}\,, \end{aligned} \tag{2.5}$$

where k is the number of bits and M is the 'level', $M = 2^k$ [4].

2.3 DS-CDMA Model

CDMA operates on the principle that signals can be decoded if they are orthogonal to each other. The user's signal is decoded by mixing the received signals with orthogonal signature waveforms. The auto-correlation will allow the desired waveform to pass; yet as the signature waveforms are orthogonal, the cross-correlation 'cancels' components from any other user.

The basic model of a synchronous CDMA system can be given by the mathematical description of the *k*th user:

$$x_k(t) = A_k^{Tx} \sum_{i=0}^{\infty} b_k(i) s_k(t - iT), \tag{2.6a}$$

where A_k^{Tx} and s_k are the transmitted amplitude and spreading waveform of the *k*th user respectively, $b_k(i)$ is the *i*th bit of the *k*th user and T is the bit duration. The asynchronous case is described below:

$$x_k(t) = A_k^{Tx} \sum_{i=0}^{\infty} b_k(i) s_k(t - iT - \tau_k), \tag{2.6b}$$

where τ_k is the time delay. The model is shown below:

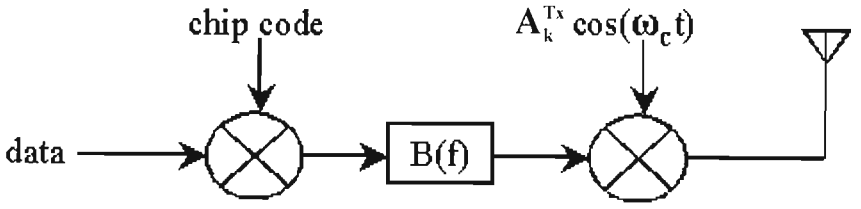


Figure 2.2: DS-CDMA transmitter

In Figure 2.2 above, B(f) represents a wave-shaping filter, and the chip code is a pseudo-noise (PN) sequence. The spreading waveform is given by:

$$s_k(t) = \sum_{n=0}^{N-1} c_k(n) \psi(t - nT_c), 0 \le t \le T_c, \tag{2.7}$$

where N is the processing gain, c_k is the *k*th users chip code, $\psi(t)$ is the chip pulse shape and T_c is the chip duration.

The synchronous received signal for the *k*th user is then given by:

$$r(t) = \sum_{k=1}^{K} A_k^{Rx} b_k s_k(t) + \sigma n(t), t \in [0, T], \quad (2.8)$$

where A_k^{Rx} is the received amplitude of the *kth* user. The noise term $n(t)$ has components due to AWGN and multiple access interference (MAI). The receiver model is shown in Figure 2.3:

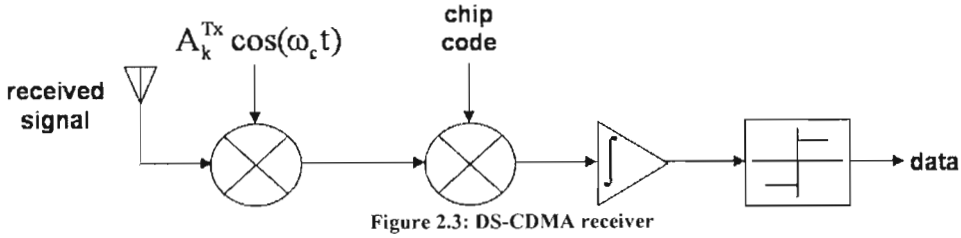


Figure 2.3: DS-CDMA receiver

Figure 2.4 below shows the waveforms for the data bit stream (BPSK), the chip sequence and the product of the two: the spread data. Note that the transitions of the data sequence coincide with a transition in the chip sequence. If the chip sequence were truly random, then the product (spread) sequence would be another random sequence with the same chip rate as the PN sequence, whereas the spread waveform in Figure 2.4 is similar to the chip sequence.

The probability of a bit error for a single-user CDMA system is:

$$P_e = Q\left(\sqrt{2E_b/N_0}\right), \quad (2.9)$$

where E_b is the signal energy per bit and $\frac{1}{2}N_0$ is the power spectral density of the noise. Figure 2.5 shows the performance for a single-user system, using a length-31 Gold code for spreading (see the following section for an outline of spreading codes).

2.4 Pseudo-Noise Sequences

The chip codes that determine the signature waveform of the users are pseudo-noise (PN) sequences, which can be generated by a series of shift registers with feedback. The basic PN sequence is the maximal length (ML) sequence. The generated sequence has a length $L = 2^N - 1$, where N is the number of shift registers employed. PN sequences have a number of properties, which are described below:

- PN sequences are 'balanced', that is the number of '1's and the number of '0's differ by 1.

- The sequences have 'runs' of '1's or '0's (a run is the number of consecutive '1's or '0's). Half of the runs have a length of 1, a quarter of the runs have length 2, an eighth

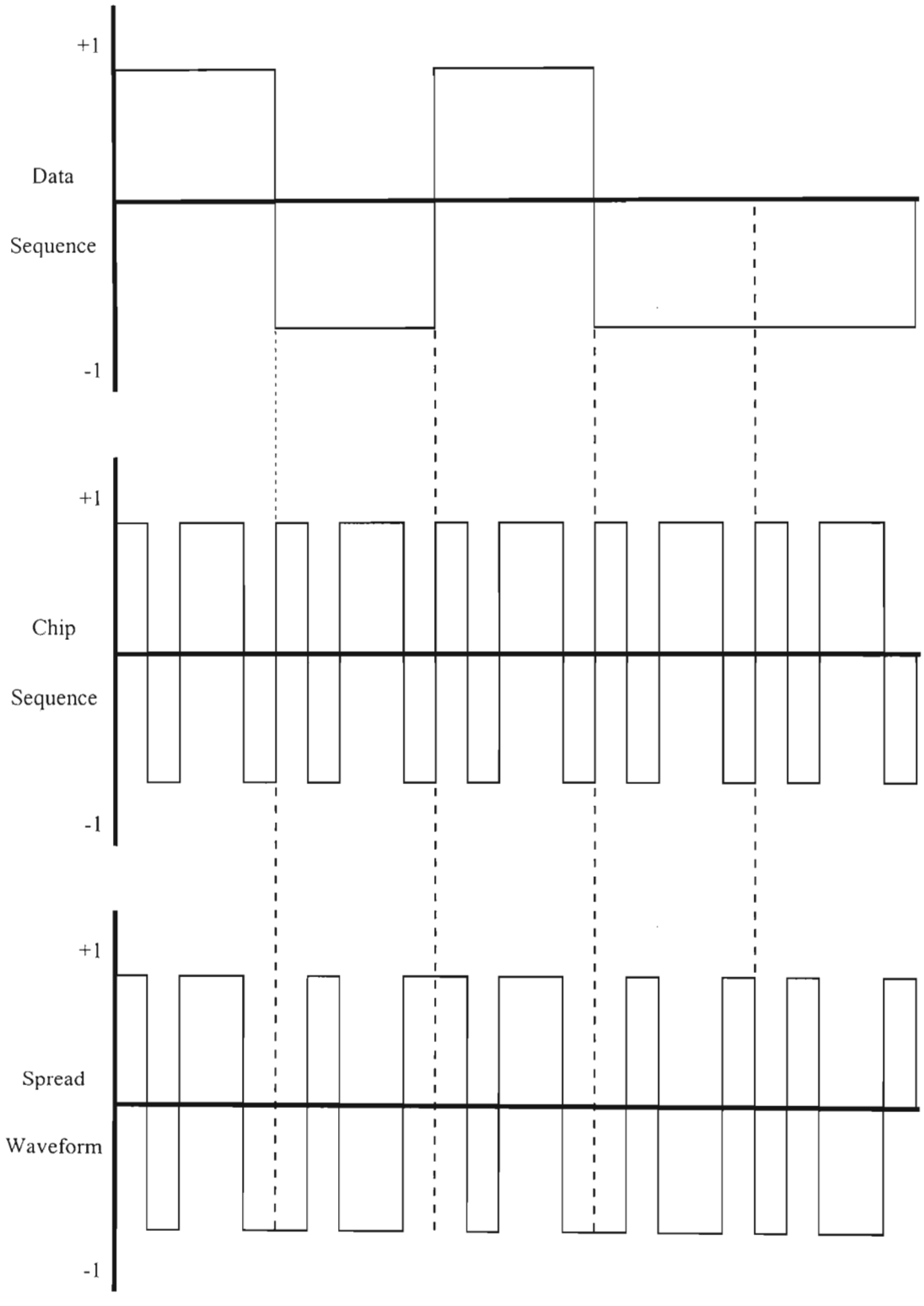


Figure 2.4: The data, chip and spread (product) waveforms

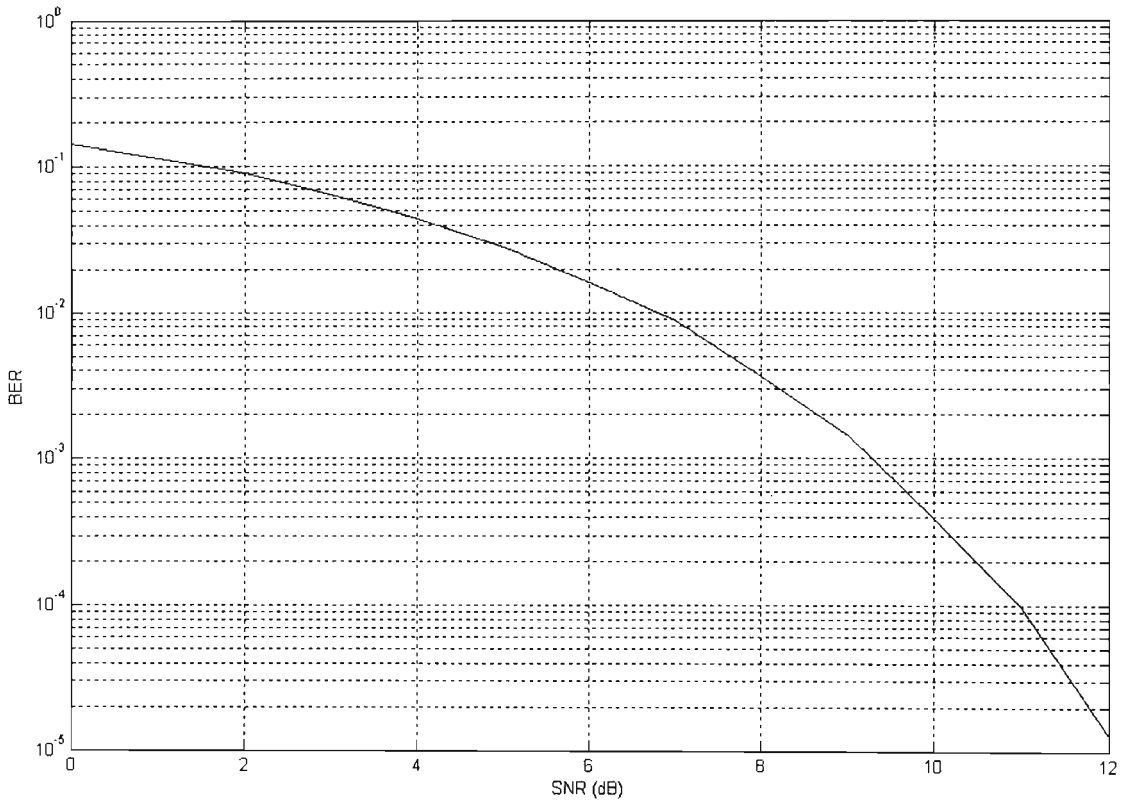


Figure 2.5 Simulated CDMA single-user performance

of the runs have length 3 and the remaining one eighth runs cannot be counted as the code is too short [2].

- PN sequences have a correlation between one another, and can be determined by the equation:

$$P = \frac{1}{L}(N_s - N_d), \tag{2.10}$$

where N_d and N_s are the number of differences and similarities, respectively, when two sequences are compared, and L is the length as defined above [2].

Other PN sequences include Walsh Codes, which are generated recursively and are used in synchronous channels and WCDMA downlinks because of their good orthogonality, and Gold Codes, which are generated by summing pairs of ML sequences and are usually used in asynchronous channels because of their good cross-correlation. They have a guaranteed 3-level cross-correlation [4], given below:

$$\{t(N)/L,\ 1/L,\ (t(N)-2)/L\}. \tag{2.11}$$

For a L-length code there are L + 2 codes in a family and

$$t(N) = \begin{cases} 1+2^{(N+1)/2}, \text{for odd } N \\ 1+2^{(N+2)/2}, \text{for even } N \end{cases} \tag{2.12}$$

Figure 2.6 shows examples of generators for a ML sequence and a gold code. It can be clearly seen that the gold code is the sum of two ML sequences. The codes produced from the above generators will be of length 31 ($2^5 - 1 = 31$).

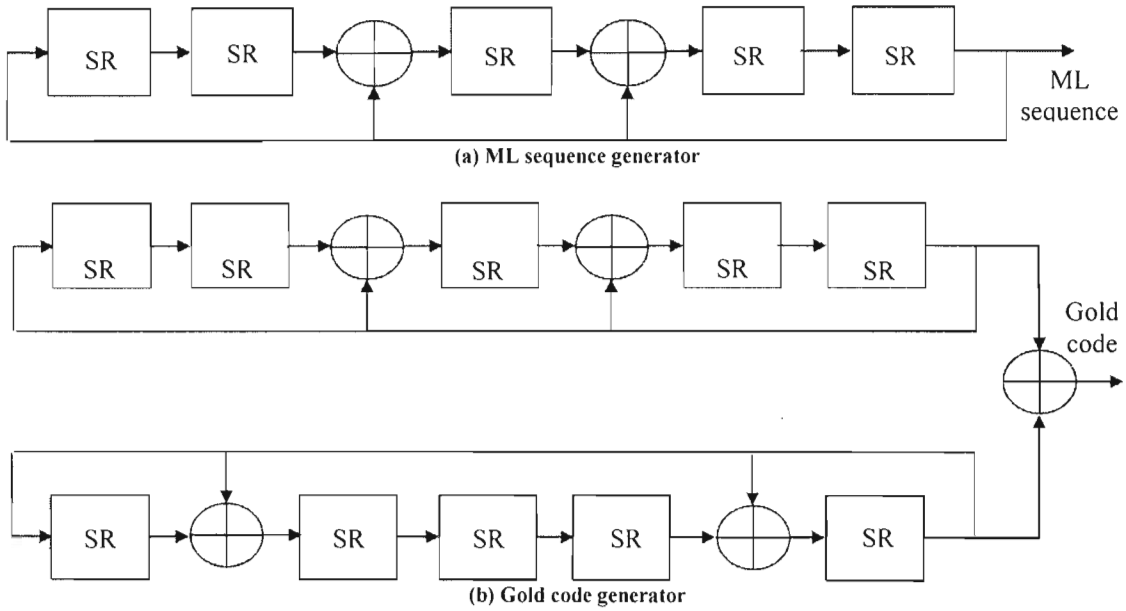


Figure 2.6: PN sequence generators

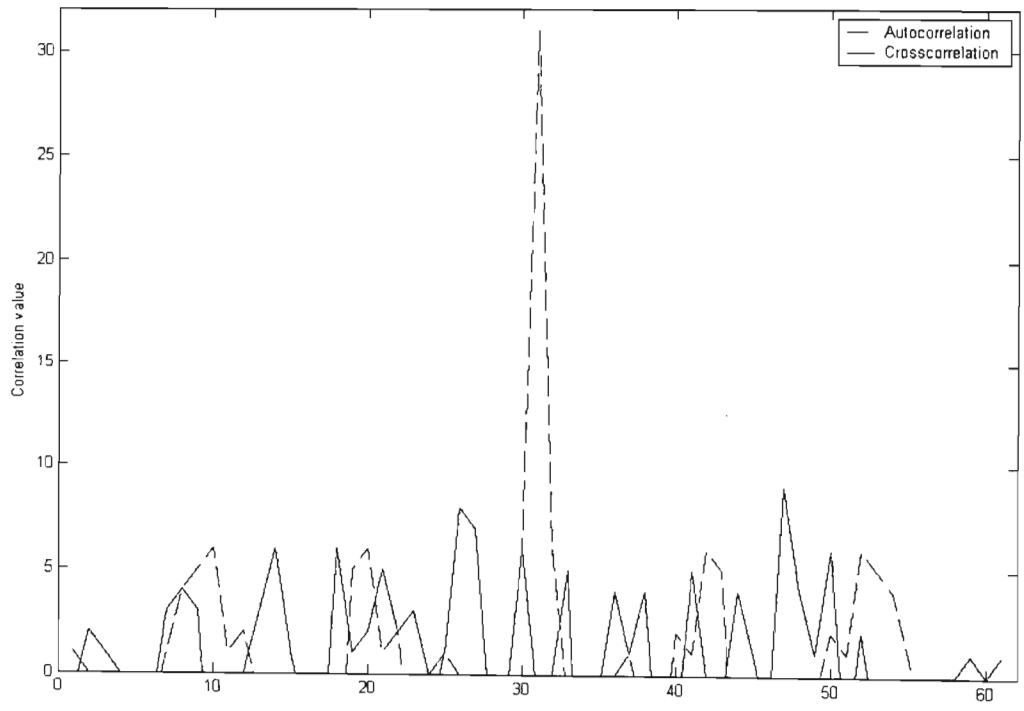


Figure 2.7: Autocorrelation and cross-correlation values

Figure 2.7 shows the autocorrelation and cross-correlation properties of pseudo-noise sequences. A length 31 Gold Code sequence was used to generate the plots in Matlab. It can be seen that they have good autocorrelation, but poor cross-correlation.

2.5 Existing CDMA Systems

As already mentioned, there are a number of CDMA systems currently in use. Here we will look briefly at the IS-95, IS-665 and CDMA 2000 protocols. An outline of the protocol is given, and the main parameters of the protocol are listed to give an idea of the bandwidths, chip rates and spreading sequences that are currently in use.

2.5.1 IS-95 (CDMA One)

This standard was first introduced in 1989, as a "Common Air Interface", which eventually became the IS-95 standard in 1993, and was then revised again in 1995 to become IS-95A. This is a dual-mode standard with an AMPS analogue mode. The IS-95B standard was completed in 1998. The IS-95A standard only uses one spreading code per channel, whereas the IS-95B standard has the ability to concatenate up to eight codes for the transmission of higher bit rates, providing a maximum data rate of 115.2 kbps. IS-95 was commercially available in the United States, Hong Kong, Singapore and Korea, and was very successful in Korea [5]. The parameters for the IS-95 standard are listed in Table 2.1 [5], and will be discussed in more detail in the following paragraphs. A more detailed description and diagrams of the various channels may be found in [5].

Table 2.1 IS-95 Parameters

Bandwidth	1.25 MHz
Chip rate	1.2288 Mcps
Frequency band uplink	869 – 894 MHz 1930 – 1980 MHz
Frequency band downlink	824 – 849 MHz 1850 – 1910 MHz
Frame length	20ms
Bit rates	Rate Set 1: 9.6, 4.8, 4.2, 1.2 Kbps Rate Set 2: 14.4, 7.2, 3.6, 1.8 Kbps IS-95B: 115.2 Kbps
Speech codec	QCELP 8 kbps EVRC 8 kbps ACELP 13 kbps
Soft handover	Yes
Power control	Uplink: Open loop and fast closed loop Downlink: Slow quality loop
No. RAKE fingers	4
Spreading codes	Walsh and long M-sequence

The channel downlink structure consists of three common (i.e. they are shared) channels and one dedicated channel (they are allocated for the use of a single user). The paging channel, the

pilot channel and the synchronisation channel are common, whilst the traffic channel is dedicated. The data that is to be transmitted over a common channel is grouped into 20ms frames, convolutionally encoded, repeated to adjust the data rate, and then interleaved. Orthogonal Walsh codes are used to spread the signal at a rate of 1.2288 Mcps. The signal is split into the I and Q channels and is spread with long PN sequences at a rate of 1.2288 Mcps prior to baseband filtering. The pilot channel is used by the mobile station for coherent demodulation, acquisition, tracking of time delays, power control measurements and as an aid for handover. The synchronisation and paging channels operate at a fixed rate of 1.2 Kbps and 9.6 Kbps, respectively [5]. Each downlink traffic channel contains one fundamental code channel and up to seven supplemental code channels, and has two different rate sets, as shown in Table 2.1. The supplemental code channels may only utilise the full data rate (9.6 or 14.4 Kbps). As each channel has a different data rate, symbol repetition is used to achieve a fixed data rate prior to interleaving. Three types of spreading codes are used: length 64 Walsh codes to separate the physical channels, a pair of long M-sequences of length $2^{15} - 1$ is used for quadrature spreading (one for the I channel and the other for the Q channel), and a long pseudo random sequence with a period $2^{42} - 1$ is used for base band data scrambling [5].

The uplink has two channels: a dedicated traffic channel and a common access channel. The traffic channel consists of a fundamental code channel and eight (0 to 7) supplemental code channels. As in the downlink, two rate sets can be supported (rate set 2 does not necessarily have to be supported), and only the full data rates are used in the supplemental code channels. Data that is transmitted on the uplink channels are grouped into 20ms frames, convolutionally encoded, block interleaved and modulated by 64-ary orthogonal modulation. The signal is spread using long PN sequences at a rate of 1.2288 Mcps prior to baseband filtering, then split into the I and Q channels and then spread with in-phase and quadrature spreading sequences [5]. The access channel is used by the mobile station to initiate a call, to respond to a paging channel message, and for location updates. The access channel only operates at a fixed data rate of 4.8 Kbps. Each access channel is associated with a paging channel, hence there can be up to seven access channels. The transmitted data is convolutionally encoded using the same generator polynomials as the downlink. For rate set 1 on the traffic channels, a rate 1/3 convolutional code is used, and a rate 1/2 is used for rate set 2. As with the downlink, there is symbol repetition, however here they are not transmitted, but masked out to save transmission power. In the access channel however, the repeated symbols are transmitted. The coded symbols are grouped into groups of six, which are used to select one of 64 Walsh codes, which are used for orthogonal modulation. Each code channel and each access channel is identified by a different phase of a pseudo random M-sequence of length 2^{42}. The in-phase and quadrature spreading is performed by the same length 2^{15} M-sequences as in the downlink channel [5].

2.5.2 IS-665 (W-CDMA)

This standard is not to be confused with the UMTS/WCDMA standard. IS-665 W-CDMA has it origins in broadband CDMA (B-CDMA) concept, which was introduced in 1989. The original bandwidth was large: 48 MHz and the chip rate was 24 Mcps. In 1991 W-CDMA was proposed and became the IS-665 standard. As it was not commercially successful, it has not been deployed except in trial systems. It was also the basis of the Japanese Core-B proposal [5].

The main parameters for IS-665 W-CDMA are shown in Table 2.2 [5]. The 64 Kbps data rate is transmitted using a ½ rate convolutional code, whilst the lower rates use symbol repetition. The Core-B proposal is an enhanced version, and offers 128 Kbps within a single code and 2 Mbps using multicode transmission. The frame length is 5ms, and interleaving can be performed over 5, 10 or 20 ms. IS-665 has open and closed loop control, the open loop control having variable step sizes, which are controlled adaptively. The use of interference cancellation is supported.

Table 2.2 IS-665 W-CDMA Parameters

Bandwidth	5, 10, 15 MHz
Basic chip rate	4.096, 8.192 and 12.288 Mcpss
Basic data rate	16, 32, 64 Kbps
Base station synchronisation	Synchronous
Frame length	5ms
Multirate / variable rate	Symbol repetition / multicode
Coherent detection	Pilot Cannel
Power control	2 Kbps time multiplexed

2.5.3 CDMA 2000

The main focus of CDMA 2000 is to provide 144 Kbps and 384 Kbps in a 5 MHz bandwidth for vehicular and pedestrian environments respectively, and 2048 Kbps for the indoor environment. Table 2.3 [5] lists the parameters for the CDMA 2000 protocol, and the following paragraphs will give an overview of the channel structure. Refer to [5] for a more in-depth description of this protocol [5].

The uplink has four dedicated channels: the fundamental and supplemental channels for user data and the control and pilot channels. The control channel has a frame length of 5 or 20ms, and carries information such as measurement data, and the pilot channel is used as a reference for coherent detection, and also carries time multiplexed power control symbols. These dedicated channels are separated by Walsh codes [5]. There is also a reverse access channel (R-ACH), which is used for initial access, and a reverse common control channel (R- CCCH), which is used for fast packet access. Both of these are common channels. The fundamental channel carries voice, signalling and low rate data, and supports rates of 9.6 and 14.4 Kbps and

their subrates (rate sets 1 and 2 of IS-95), and it always operates in the soft handover mode. It does not operate in a scheduled manner, which reduces delays and processing loads. The uplink supports up to two supplemental channels, which provide high data rates. In the uplink, user separation is performed by different phase shifts of a 241 length M-sequence, and channel separation is performed using variable spreading factor Walsh sequences [5].

Table 2.3 CDMA 2000 Parameters

Bandwidth	1.25, 5, 10, 15, 20 MHz
Downlink RF Structure	Direct spread or multicarrier
Chip rate	1.2288/3.6864/7.3728/11.0593/14.7456 Mcps for direct spread n x 1.2288 Mcps (n=1,3,6,9,12) for multicarrier
Frame length	20ms for data and control / 5ms for control information on the fundamental and dedicated control channel
Spreading modulation	Balanced QPSK (downlink) Dual-channel QPSK (uplink) Complex spreading circuit
Data modulation	QPSK (downlink) BPSK (uplink)
Coherent detection	Pilot time multiplexed with PC and EIB (uplink) Common continuous pilot channel and auxiliary pilot (downlink)
Channel multiplexing (uplink)	Control, pilot, fundamental and supplemental code multiplexed I&Q multiplexing for data and control channels
Multirate	Variable spreading and multicode
Spreading factors	4 – 256
Power control	Open loop and 800Hz fast closed loop
Spreading (downlink)	Variable length Walsh sequences for channel separation, M-sequence 2^{15} (same sequence with time shift utilised in different cells, different sequence in I&Q channel)
Spreading (uplink)	Variable length orthogonal sequences for channel separation, M-sequence 2^{15} (same for all users different sequences in I&Q channels), M-sequence $2^{41} - 1$ for user separation (different time shifts for different users)
Handover	Soft Handover Interfrequency handover

The downlink has three dedicated and three common control channels. As with the uplink, the fundamental and supplemental channels carry user data and dedicated control channel carries the control messages, containing power control bits and rate information. Mobile stations use the synchronisation channel to acquire initial time synchronisation. One or more paging channels are used for paging the mobiles. The reference signal for coherent detection, cell acquisition and handover is provided for by the pilot channel, and is similar to the IS-95 standard in that it is comprised of a long PN-sequence and Walsh code [5].

2.6 Conclusion

This chapter has presented an overview of multi-access and spread spectrum communications. The operation and performance of DS-CDMA was described. Three existing CDMA standards were described. Whilst these standards offer power-control to minimise the near-far effect, they do not offer a solution to all the problems encountered in multiple access channels. Chapter 3 will discuss multiuser detection, which further suppresses interference due to multiple users.

CHAPTER 3 MULTIUSER DETECTION

This chapter presents an overview of multiuser detection for DS-CDMA communication systems. The need for multiuser detection is discussed, and various techniques are described. The focus is on the minimum output energy (MOE) blind multiuser detector (MUD), as it is the most popular of the blind detectors [6] and is required for the proposed detector. The advantages of blind detection over other MUD techniques are discussed. The performance of the linear multiuser detectors is compared.

3.1 Conventional Detector

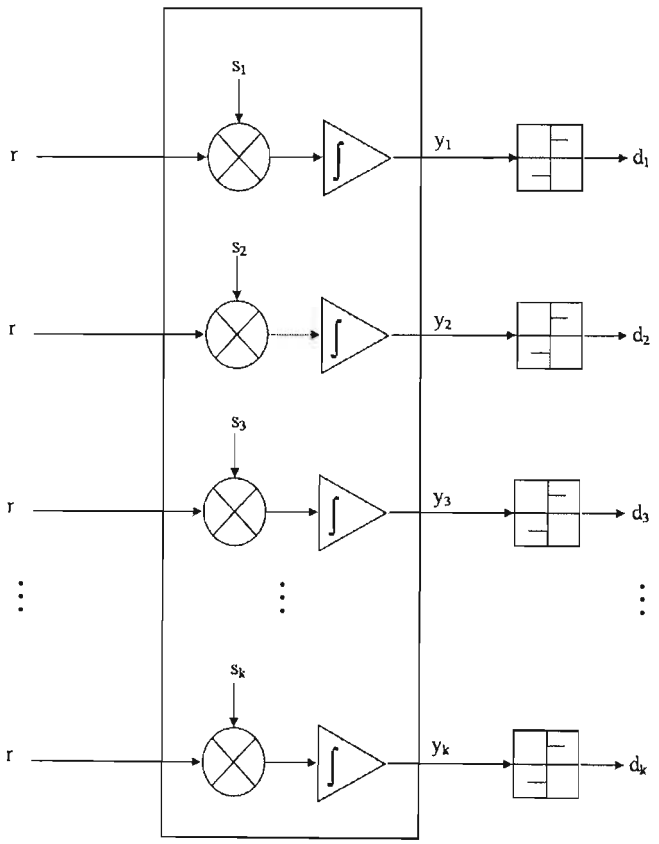


Figure 3.1: Conventional CDMA detector

Conventional CDMA detectors are just a bank of single-user detectors (chip matched filters), as shown in Figure 3.1. The received signal is given by Equation 2.8:

$$r(t) = \sum_{k=1}^{K} A_k^{Rx} b_k s_k(t) + \sigma_n(t), t \in [0, T], \tag{2.8}$$

and the output of the matched filter is given by:

$$y_k(t) = \int \mathrm{Re}\{r(t)e^{-j\varphi_k}\} s_k(t - iT - \tau_k) dt, \tag{3.1}$$

where φ_k and τ_k are the phase and time delay respectively [1, 5, 6]. The output of the matched filter is passed through a decision, and the estimated bits are then given by

$$\hat{d}_k(i) = \text{sgn}(y_k(i)), \tag{3.2}$$

Equation 2.8 can be re-written using matrix representation to model the conventional multiuser system:

$$\mathbf{r}(i) = \mathbf{SAb}(i) + \mathbf{n}(i), \tag{3.3}$$

where $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_k]$, $\mathbf{A} = \text{diag}([\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_k])$, $\mathbf{b} = [b_1, b_2, \cdots, b_k]^T$ and $\mathbf{s}_k = [c_k(1), c_k(2), \cdots, c_k(N)]^T$. The AWGN vector n(i) has a covariance matrix of $\sigma^2\mathbf{I}_N$.

As mentioned in Section 2.3, the noise term, n(t), is comprised of AWGN and MAI. The MAI component of the noise term can be given by:

$$\text{MAI} = \sum_{j \neq k}^{K} x_j \int_0^T s_k(t) s_j(t) dt, \tag{3.4}$$

where x_j is the jth user's signal, s_k is the desired user's spreading sequence and s_j is the jth user's spreading sequence, and $j \neq k$.

Due to the effect of MAI, as the number of users increase in the channel, the performance of the system is degraded. Figure 3.2 shows the effect the number of users has on the performance of a single-user detector in an AWGN channel for a SNR of 10 dB, using a length-31 spreading code. As can be seen, the semi-log graph shows a straight line, hence the number of errors increases exponentially with the number of users.

Figure 3.3 shows the effect the length of the spreading code has on the performance of the conventional CDMA detector for an AWGN channel with a SNR of 10 dB. As can be seen, the performance is improved for an increase in the code length, due to the increase in the processing gain and the spectral efficiency, as calculated by Equations 2.4 and 2.5, respectively. There is a disadvantage of using very long PN sequences is that they are not orthogonal [6], hence there will be an increased cross-correlation, leading to an increase in MAI.

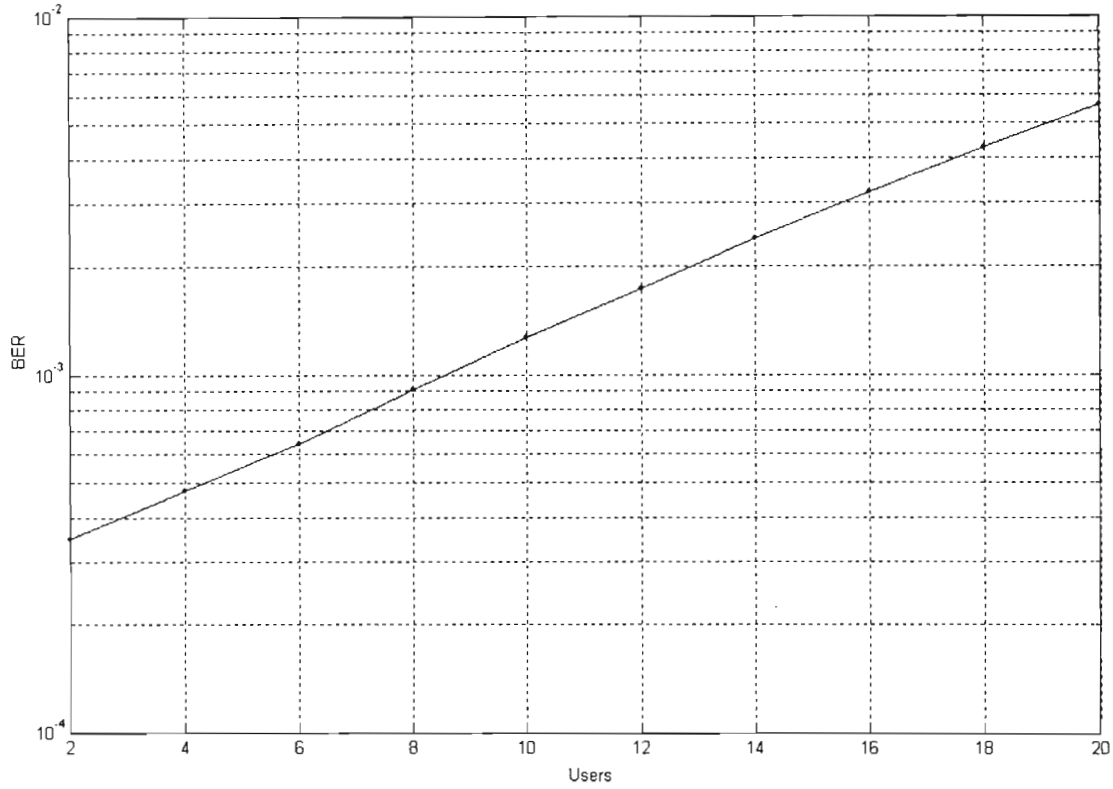


Figure 3.2: The effect of channel population on performance, SNR = 10dB.

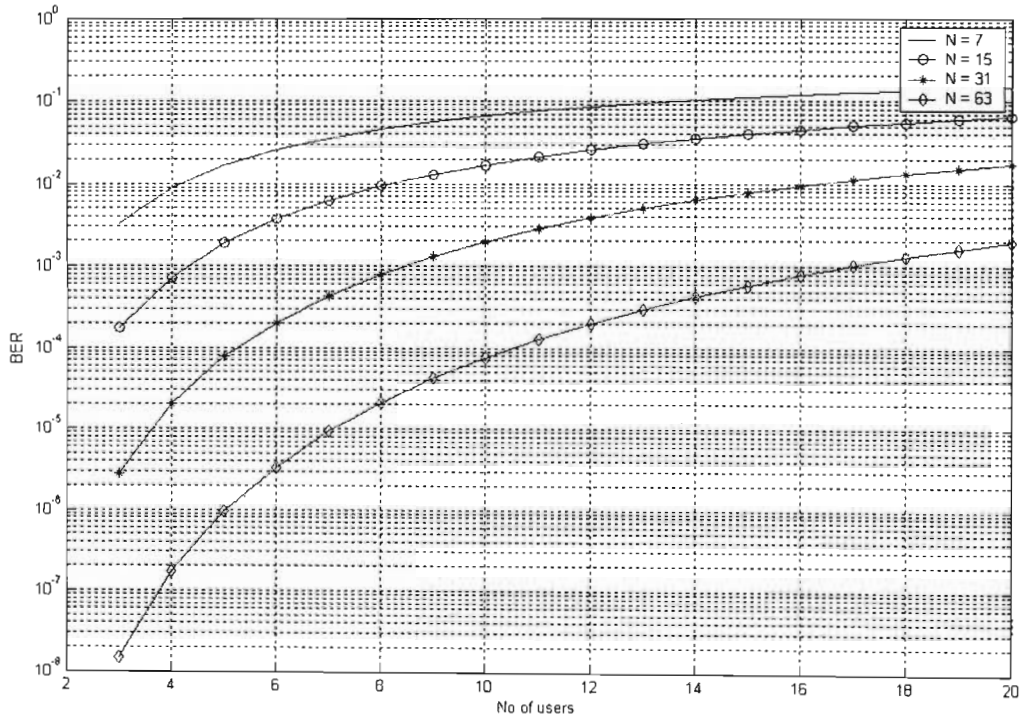


Figure 3.3: Theoretical effect of the pseudo-code length on performance, SNR=10dB

3.2 The Multiuser Channel

As can be seen from Figure 3.2, as the number of users increase, the performance is degraded. The presence of additional users creates problems that need to be taken into account when detecting data. Of these problems, MAI has already been discussed above, and multipaths and the near-far problem will be looked at in Sections 3.2.1 and 3.2.2, respectively.

3.2.1 Multipaths

Signals can take different 'routes' to their 'destination'. Each route will have its own phase shifts and delays, hence different components of the signal will be asynchronous. For n multipath components, each interfering user will create n interfering signals, hence the detector will have to eliminate $n(K\text{-}1)$ signals. The desired user will also have n components, which will have to be combined. The multipaths were simulated by generating random values with, magnitude less than half the main signal strength, and shifting the "bits" in the matrix form relative to the main signal to account for propagation delay. Figure 3.4 below shows the effect that the number of multipaths, which have different strengths, has on the performance of a CDMA system.

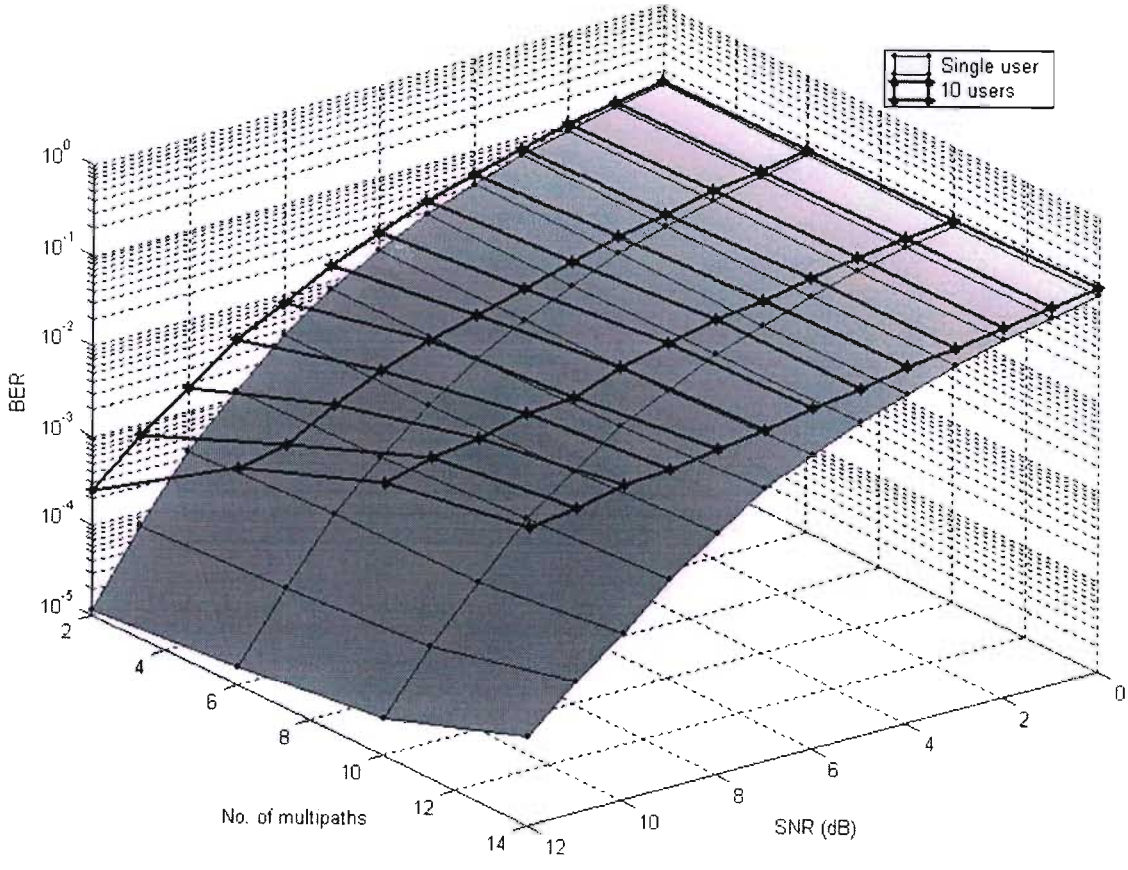


Figure 3.4 Simulated effect of multipaths

As can be seen from Figure 3.4, the performance is degraded with a larger number of multipaths. As the number of users increases the effect of the number of multipaths becomes more pronounced, as the number of interfering components increases.

3.2.2 The Near-Far Problem

Signals will be received with different powers. If the desired signal is further from the receiver than the interfering user(s), or they are transmitting with a greater power than the desired user, the signal at the receiver is in danger of being 'drowned-out' by the interfering signals. In Figure 3.5, the signal strength of the interfering users is the ratio of the interfering signal strength A_k to that of the normalized desired signal strength A_1, and is given by $10\log\frac{A_k^2}{A_1^2}$, but as A_1 is normalized, i.e. A_1=1, this becomes $10\log A_k^2 = 20\log A_k$. As with the multipath length, it can be seen that the effect on the system is greater when there are more users present. As the number of users increases, there is more interference with the desired user's signal, therefore there will be a degradation in performance. For the single user case, there will be no effect, as there are no other users to create interference. All of the 10 users in Figure 3.5 have greater power than the desired user.

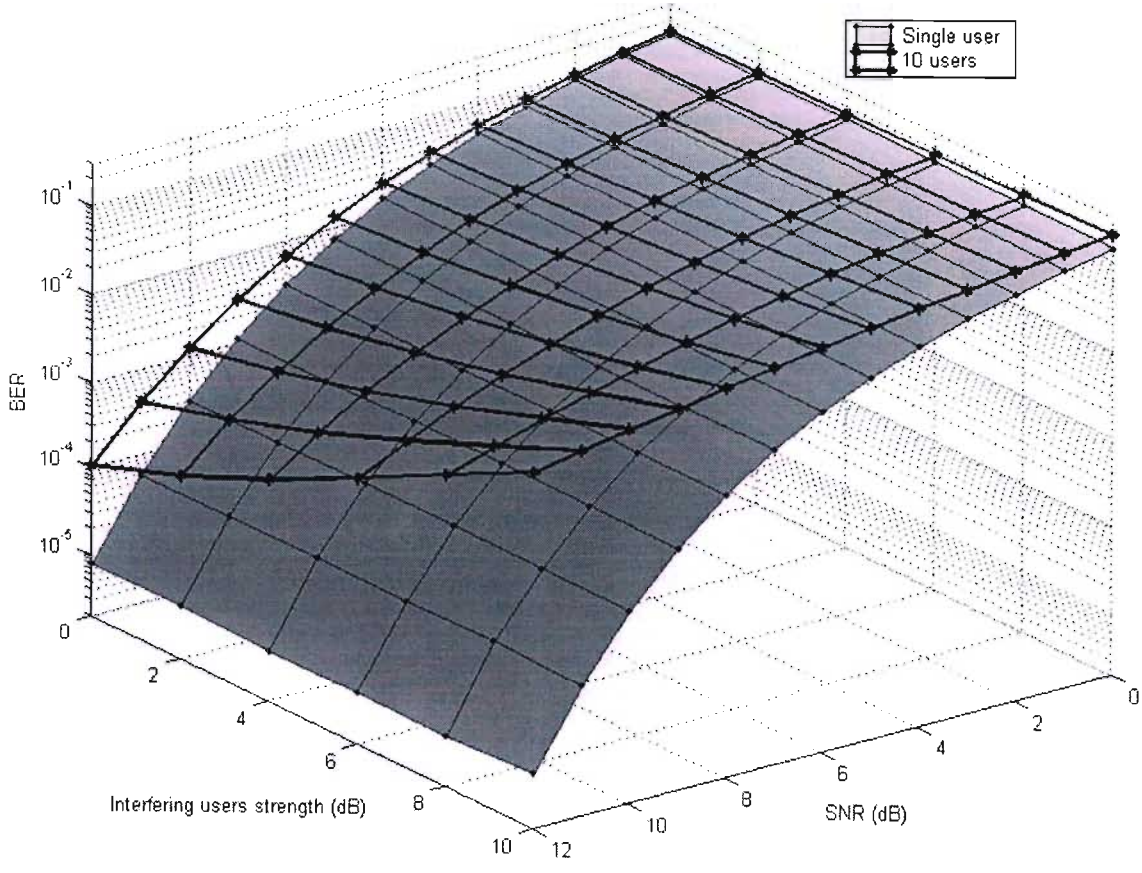


Figure 3.5 Simulated near-far effect

3.3 Optimal MUD

Optimal multiuser detection is based on maximum likelihood sequence estimation (MLSE). The maximization of a cost function leads to the optimum decoding of all users [6]. The most likely hypothesis $\hat{\mathbf{b}}^* = (\hat{b}_1^*, \cdots, \hat{b}_k^*)$ is selected for the given observations, which is equivalent to selecting the noise realization with minimum energy (the Euclidean distance between the received vector and the set of possible noiseless received vectors is minimized) [1, 7-11]:

$$\begin{aligned} \hat{b}^* &= \arg \min_{b \in \{-1,1\}^K} \int_0^T \left[r(t) - \sum_{k=1}^{K} b_k s_k(t) \right]^2 dt \\ &= \arg \max_{b \in \{-1,1\}^K} 2\mathbf{y}^{\mathrm{T}}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathbf{R}\mathbf{b} \end{aligned} \tag{3.5}$$

where $\mathbf{y}$ is the transmitted bits, $\mathbf{b}$ is decision based on $\mathbf{y}$, and $\mathbf{R}$ is the non-negative matrix of cross-correlations between waveforms:

$$\mathbf{R}_{ij} = \int_0^T s_i(t) s_j(t) dt \,. \tag{3.6}$$

The maximization of the cost function (3.5) is a combinatorial optimization problem, meaning that the possible arguments comprise a finite set. Hence it is possible to use an exhaustive search i.e. perform calculations for all possible arguments and choose the argument that maximizes the algorithm. The computational complexity of optimal detection increases exponentially with the number of users [6,7].

3.4 Decorrelating Receiver

This method applies the inverse of the correlation matrix to the soft output of the conventional detector, and is analogous to zero-forcing equalizers. The correlation matrix is defined by Equation 3.6. The output of the decorrelating detector is given by [1, 5-9, 11-14]:

$$\hat{\mathbf{d}}_{\mathrm{CORR}}(i) = \mathrm{sgn}(\mathbf{R}^{-1}\mathbf{y}(i)), \tag{3.7}$$

where $\mathbf{y}(i)$ is the output of the matched filter and $\mathbf{R}$ is the KxK correlation matrix, given by

$$\mathbf{R} = \begin{bmatrix} 1 & \rho_{2,1} & \rho_{3,1} & \cdots & \rho_{K,1} \\ \rho_{1,2} & 1 & \rho_{3,2} & \cdots & \rho_{K,2} \\ \rho_{1,3} & \rho_{2,3} & 1 & \cdots & \rho_{K,3} \\ \vdots & \vdots & \vdots & 1 & \vdots \\ \rho_{1,K} & \rho_{2,K} & \rho_{3,K} & \cdots & 1 \end{bmatrix}.$$

The advantages of this detector is that it eliminates MAI and it does not require knowledge of the users' received amplitudes, hence its probability of error is independent of signal energies, rendering it near-far resistant. The detection of each of the users can be performed independently of the other users. However, the detector does not result in optimum decisions, and tends to enhance noise [1, 5-9, 11-14]. The matrix inversion may become computationally intensive for a large number of users, or when long spreading codes are used.

3.5 MMSE MUD

Minimum mean squared error (MMSE) receivers take into account background noise and makes use of received signal powers. As the names suggests, the mean squared error between the transmitted bit and the decision variable is minimized i.e. $\min\left(\mathrm{E}\left[b_k - \langle s_k, y\rangle^2\right]\right)$ [1, 6-7, 13-18]. The linear transform that achieves MMSE is given by:

$$\mathbf{L}_{\mathrm{MMSE}} = \left[\mathbf{R} + (\sigma' \mathbf{A}^{-1})^2\right]^{-1}. \tag{3.8}$$

The bit decision is then given by:

$$\hat{\mathbf{d}}_{\mathrm{MMSE}} = \mathrm{sgn}(\mathbf{L}_{\mathrm{MMSE}}\mathbf{y}). \tag{3.9}$$

As the MMSE receiver takes into account the background noise, the noise enhancement problem experienced by the decorrelating detector is circumvented. The BER performance is thus superior to that of the decorrelating receiver, and as the background noise goes to zero, the MMSE detector approaches the decorrelating receiver. The disadvantages are that estimation of the received power levels, have to be estimated, so erroneous estimations may degrade performance. As performance is dependant on the signal power levels, the receiver is not as near-far resistant as the decorrelating detector [1, 6-7, 13-17]. Another advantage of the MMSE receiver is that it can be implemented as an adaptive receiver, however it will need training as discussed in Section 3.6.

3.6 Training Based MMSE Receiver

There are two main adaptive implementations of the MMSE receiver: the Recursive Least Squares (RLS) and the Least Mean Squares (LMS). The RLS receiver update equations are as follows [7, 9]:

$$w_k[i+1] = w_k[i] + P[i]r[i]\varepsilon_k[i] \quad \text{and}$$
$$P[i+1] = \rho^{-1}P[i] - \frac{P[i]r[i]r[i]^T P[i}{\rho + r[i]^T P[i]r[i]},$$

where $w_k[i]$ is the cost function, $r[i]$ denotes the received bits and $\varepsilon_k[i] = b_k[i] - w_k[i]r[i]$ is the estimation error for the kth user. The LMS update equation is [7, 9]:

$$w_k[i+1] = w_k[i] + \mu\varepsilon_k[i]r[i],$$

where μ is the step size.

As can be seen, the RLS detector has a higher computational complexity than the LMS detector; however it has a faster convergence and lower steady state error [1, 6, 7, 9]. The training sequence is known at the receiver, and is transmitted before the message bits. As the received training sequence is known, it can be used to initialise the filter weights, which are updated by either the RLS or LMS equations.

3.7 Blind MUD

There are two main methods for blind detection: the minimum output energy (MOE), which is an adaptive algorithm, and a subspace-based computation. The MOE method is more suited for an iterative detector as it allows for the estimation of channel parameters, which are used to calculate priors in the iterative detectors.

In this section the equations for a blind adaptive receiver using the minimum output energy criterion are derived. These detectors have the advantage of not using a training sequence, hence their name, yet they approximate the MMSE receivers.

3.7.1 MOE Detector

Linear multiuser detectors can be characterised by the waveform c_1, which replaces the spreading sequence s_1, such that [1,17-19]

$$\hat{b} = \mathrm{sgn}\left(\langle y, c_1 \rangle\right), \tag{3.10}$$

where the inner product notation denotes $\langle x, y \rangle = \int_0^T x(t)y(t)dt$.

The canonical representation for c_1 is:

$$c_1 = s_1 + x_1, \tag{3.11}$$

where x_1 is orthogonal to the spreading sequence s_1, so:

$$\langle s_1, x_1 \rangle = 0. \tag{3.12}$$

Every linear multiuser detector can be expressed in the form of Equations (3.11) and (3.12) as the set of signals c_1 that can be written in this form satisfy

$$\langle s_1, c_1 \rangle = \|s_1\|^2 = 1. \tag{3.13}$$

Every linear transformation for multiuser detection can be characterised by its corresponding orthogonal signal x_1. Given a linear transformation d_1 the orthogonal component is given by

$$x_1 = \frac{1}{\langle s_1, d_1 \rangle} d_1 - s_1. \tag{3.14}$$

A measure of performance is the signal-to-interference ratio (SIR) at the output of the linear transformation. This is a useful measure of performance in situations where the background noise is not negligible with respect to the multi-access interference (MAI). A linear detector in canonical form has the following SIR [1, 19]:

$$\mathrm{SIR} = \frac{A_1^2 \left(\langle c_1, s_1 \rangle\right)^2}{\sigma^2 \|c_1\|^2 + \sum_{k=2}^{K} A_k^2 \left(\langle c_1, s_1 \rangle\right)^2} \tag{3.15}$$

$$= \left(\frac{\sigma^2}{A_1^2} \left(1 + \|x_1\|^2\right) + \sum_{k=2}^{K} \frac{A_k^2}{A_1^2} \left(\langle s_1 + x_1, s_k \rangle\right)^2 \right)^{-1}. \tag{3.16}$$

The bit-error-rate (BER) of the linear detector defined by Equation (3.10) is given by [1, 19]:

$$P_1 = 2^{1-K} \sum_{d_2 \in \{-1,1\}} \cdots \sum_{d_K \in \{-1,1\}} Q\left(\frac{A_1 \langle c_1, s_1 \rangle + \sum A_k d_k \langle c_1, s_k \rangle}{\sigma \|c_1\|} \right). \tag{3.17}$$

In the high SNR region ($\sigma \rightarrow 0$), the BER is dominated by the largest term in Equation (3.16) and is determined by the asymptotic multiuser efficiency [1, 19]:

$$\eta_1 = \frac{\max^2\left[0{,}1 - \sum_{k=2}^{K} \frac{A_k}{A_1}\left|\langle s_k, s_1\rangle + \langle s_k, x_1\rangle\right|\right]}{1+\|x_1\|^2}. \tag{3.18}$$

The minimum asymptotic efficiency over all A_k/A_1, k = 2,...K is called the near-far resistance of the detector. The decorrelating detector is the only detector independent of A_k/A_1, k = 2,...,K that has nonzero near-far resistance, equal to $\eta_1 = \frac{1}{1+\|x_1\|^2}$ where x_1 satisfies $\langle s_k, x_1\rangle = -\langle s_k, s_1\rangle$, $k = 2,\cdots,K$ in addition to being orthogonal to s_1.

Here we consider the linear detector in canonical form that minimizes the mean output energy $E\left[\left(\langle y, s_1 + x_1\rangle\right)^2\right]$ over all x_1 orthogonal to s_1 when the input y_1 is given by $y(t) = \sum_{k=1}^{K} A_k b_k s_k(t) + \sigma n(t)$, $t \in [0,T]$. The terminology "output energy" is referring to the variance of the correlator output at time T, rather than the energy of the correlator waveform. It is important to restrict the detector to be in canonical form, otherwise the output energy is minimized with $c_1 = 0$.

The orthogonal component of the MMSE linear detector satisfies a useful property in that it maximizes the output SIR. So x_1 must be chosen in order to maximize Equation (3.16). The numerator in Equation (3.15) is not dependant on x_1, which means maximizing Equation (3.15) is equivalent to minimizing the sum of the numerator and denominator therein, which is the sum of the variance of the desired signal, and the variance of the component due to the background noise and multi-access interference [1, 19]. This sum is the variance of the output of the linear transformation:

$$\mathrm{MOE}(x_1) = \mathrm{E}\left[\left(\langle y, s_1 + x_1\rangle\right)^2\right]. \tag{3.19}$$

It is proven in [1] and [19] that the linear minimum output variance detector is the linear MMSE multiuser detector, and that the function MOE(x_1) is strictly convex over the set of signals orthogonal to s_1.

3.7.2 Blind Adaptive Algorithm

The method taken to self-tune the detector is stochastic gradient descent of a convex penalty function. For blind detection the convex penalty is the output variance. If the linear transformation is in canonical form, and the output variance is minimized with respect to the component orthogonal to the desired user's signature waveform, then the solution is the MMSE linear transformation.

The projection of the gradient of the output energy MOE(x_1) onto the linear subspace orthogonal to s_1 needs to be found, in order for the orthogonality condition Equation (3.12) to be satisfied at each step of the algorithm. The steepest descent line along the subspace orthogonal to s_1 is the projection of the gradient on that subspace. Therefore first take the unconstrained gradient of Equation (3.19) which lies in the same direction as the observed signal:

$$\nabla \text{MOE} = 2\langle y, s_1 + x_1 \rangle y . \tag{3.20}$$

The component in Equation (3.20) orthogonal to s_1 is a scaled version of the component of y orthogonal to s_1:

$$y - \langle y, s_1 \rangle s_1 .$$

The projected gradient orthogonal to s_1 is then:

$$2\langle y, s_1 + x_1 \rangle \left[y - \langle y, s_1 \rangle s_1 \right]. \tag{3.21}$$

Let the responses of the matched filters for s_1 and $s_1 + x_1[i\text{-}1]$ be denoted by

$$\begin{aligned} Z_{\text{MF}}[i] &= \langle y[i], s_1 \rangle, \\ Z[i] &= \langle y[i], s_1 + x_1[i-1] \rangle \end{aligned}$$

respectively. Then from Equation (3.21) the stochastic gradient adaptation rule is:

$$x_1[i] = x_1[i-1] - \mu Z[i]\left(y[i] - Z_{\text{MF}}[i] s_1\right), \tag{3.22}$$

which is shown in Figure 3.6. Without information about the interfering signature waveforms and for ease of computation, the initial condition for Equation (3.22) is chosen to be $x_1[i] = 0$. Equation (3.22) can be simplified to:

$$x_1[i] = x_1[i-1] - \mu (y[i] s_1)\left(y[i] - (y[i] s_1) s_1\right). \tag{3.23}$$

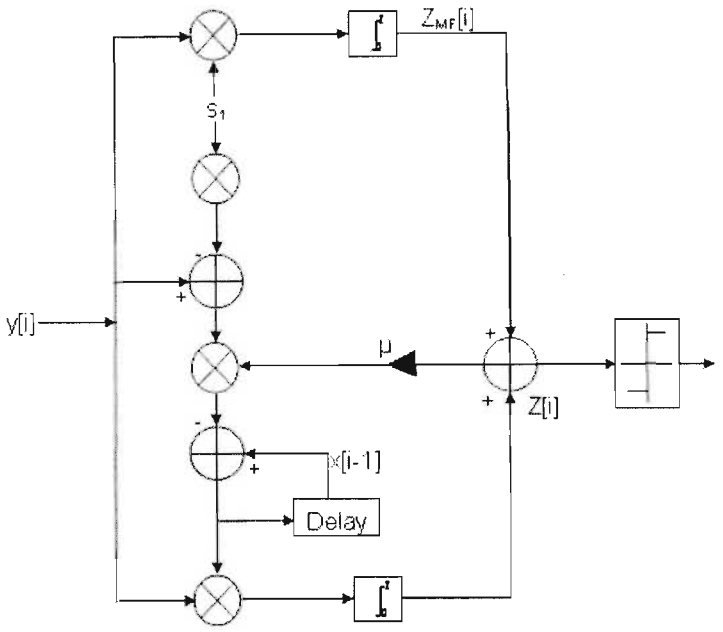


Figure 3.6: Blind adaptive receiver [1]

The adaptive algorithm in Equation (3.22) converges to the linear MMSE detector only using the same information as that of a single-user matched filter: the desired user's signature waveform and timing. In the practical implementation in Figure 3.6, finite-precision round-off error may have a cumulative effect that drives the updates outside the orthogonal subspace. This can be avoided by occasionally replacing the update $x_1[i]$ by its orthogonal projection $x_1[i]-\langle x_1[i], s_1\rangle s_1$. For the sake of reducing computational complexity and improving convergence speed, it is desirable to use the vector space with the lowest possible dimension that contains the desired and interfering signals. The more complex the recursions are, the higher the convergence speed will be. The blind detector can be implemented in asynchronous channels.

3.8 Comparison of linear MUDs

This section summarises the information requirements and performance characteristics of the linear detectors described in the sections above. The optimal detector is ignored due to the impracticality of its implementation for the purposes of this thesis.

3.8.1 Detector Requirements

Table 3.1 [1] shows the requirements for various linear multiuser detectors. As can be seen the MMSE detector requires knowledge of the timing for all the user signatures and their signature waveforms, as well as the amplitudes and noise levels. The decorrelating detector does not require information about the noise levels or amplitudes, which makes it near-far resistant, as

described in Section 3.4. The blind detector has the same information requirements as a single-user detector, but approximates the performance of a MMSE multiuser detector.

Table 3.1: Knowledge requirements for linear detectors

	Single user	Decorrelating	MMSE	Blind
Signature waveform (DU)	✓	✓	✓	✓
Timing (DU)	✓	✓	✓	✓
Amplitudes			✓	
Noise levels			✓	
Signature waveforms (IU)		✓	✓	
Timing (IU)		✓	✓	

DU – Desired user, IU – Interfering users

3.8.2 Detector Performance

The desirable measure of performance in multiuser communications is usually the probability of a bit being in error, or the bit error rate (BER). The benchmark for this measure of performance in multiuser systems is the probability of a bit error for a single-user receiver without any interfering users on the channel. This provides a lower bound for the BER performance of multiuser detectors [20]. The probability of a bit being in error for a single-user system in the absence of other users in the channel is given by:

$$P_k(\gamma_k) = Q(\sqrt{2\gamma_k}). \tag{3.24}$$

where $\gamma_k = E_k/N_0$, E_k is the signal energy per bit and $\frac{1}{2}N_0$ is the power spectral density of the AWGN [20].The Q function in Equation 3.24 can be defined as:

$$Q(x) = \frac{1}{2} erfc\left(\frac{x}{\sqrt{2}}\right). \tag{3.24}$$

The probability of error can thus be re-written as

$$P_k(\gamma_k) = \frac{erfc(\gamma_k)}{2}. \tag{3.26}$$

Let us now consider the probability of a bit error for the conventional single-user (matched filter) detector. The probability of a bit error for user k, which is conditional on a sequence of bits from the other users, is [20]

$$P_k(\mathbf{b}_i) = Q\left(\sqrt{2\left[\sqrt{E_k} + \sum_{\substack{j=1 \\ j\neq k}}^{K} \sqrt{E_k} b_j(1)\rho_{jk}(0)\right] \Big/ N_0}\right). \tag{3.27}$$

Using Equation 3.25, the probability of Equation 3.27 becomes

$$P_k(\mathbf{b}_i) = \frac{1}{2} erfc\left(\left[\sqrt{E_k} + \sum_{\substack{j=1 \\ j\neq k}}^{K} \sqrt{E_k} b_j(1)\rho_{jk}(0)\right] \Big/ N_0\right). \tag{3.28}$$

From this, the average probability can be determined as [20]

$$P_k = \left(\frac{1}{2}\right)^{K-1} \sum_{i=1}^{2^{K-1}} P_k(\mathbf{b}_i). \tag{3.29}$$

The probability P_k will be dominated by the smallest argument in the Q function, which will result in a SNR of [20]

$$(\mathrm{SNR})_{\min} = \left[\sqrt{E_k} + \sum_{\substack{j=1 \\ j\neq k}}^{K} \sqrt{E_k} \left|\rho_{jk}(0)\right|\right]^2 \Big/ N_0. \tag{3.30}$$

The bounds on performance are then [20]

$$\left(\frac{1}{2}\right)^{K-1} Q\left(\sqrt{2(\mathrm{SNR})_{\min}}\right) < P_k < Q\left(\sqrt{2(\mathrm{SNR})_{\min}}\right). \tag{3.30a}$$

$$\left(\frac{1}{2}\right)^{K} erfc\left(\sqrt{(\mathrm{SNR})_{\min}}\right) < P_k < \left(\frac{1}{2}\right) erfc\left(\sqrt{(\mathrm{SNR})_{\min}}\right). \tag{3.30b}$$

The probability of error for the decorrelator is the same as the single user case, except for a noise enhancement factor $(R^{-1})_{j,j}$ [21]:

$$P_k = Q\left(\sqrt{\frac{w_{j,j}}{(R^{-1})_{j,j} N_0}}\right) \tag{3.32a}$$

$$= \frac{1}{2} erfc\left(\sqrt{\frac{w_{j,j}}{2(R^{-1})_{j,j} N_0}}\right) \tag{3.32b}$$

where $j=(i-1)K+k$, and $w_{j,j}$ is the square root of the user energy. As all the elements of $\boldsymbol{R} \leq 1$, it follows that $(R^{-1})_{i,i} > 1$ [21]. It is difficult to find general statistics for $\boldsymbol{R}$, hence a known correlation matrix of actual user codes is generally used for calculating error probabilities.

The MMSE is similar to that of the decorrelator, where the linear transform $\boldsymbol{R}^{-1}$ is replaced by $\boldsymbol{T}^{-1} = (\boldsymbol{R} + N_0/2\boldsymbol{W}^2)^{-1}$ [21]:

$$P_k = Q\left(\sqrt{\frac{w_{j,j}}{(\mathbf{R} + N_0/2\mathbf{W}^2)^{-1}_{j,j} N_0}}\right) \tag{3.33a}$$

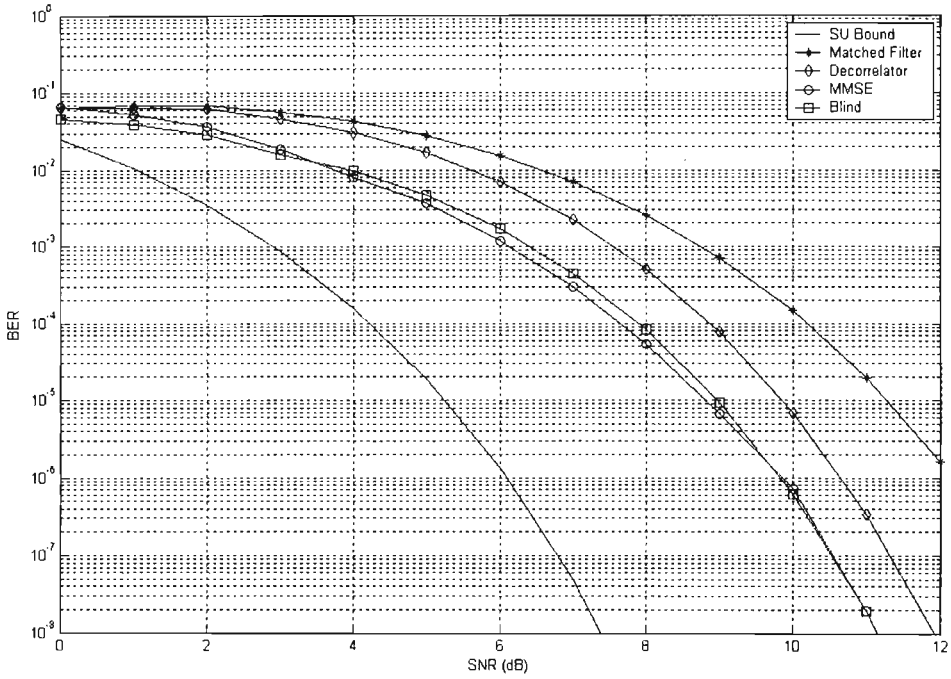
$$= \frac{1}{2} erfc\left(\sqrt{\frac{w_{j,j}}{2(\mathbf{R} + N_0/2\mathbf{W}^2)^{-1}_{j,j} N_0}}\right), \tag{3.33b}$$

where $\mathbf{W}$ is the diagonal matrix of user energies. As N_0 grows large, T tends to an identity matrix scaled by $2\mathbf{W}^2/N_0$, and thus is reduced to the conventional receiver [21]. The MMSE detector attempts to balance between removing interference and not enhancing noise. The MMSE receiver outperforms the decorrelator at low SNRs, whilst at high SNRs the performance of the decorrelator approaches that of the MMSE detector [21]. The probability for error for a blind detector is given by Equation 3.17. Figure 3.7 shows the theoretical and simulated performances for various linear MUDs in an AWGN channel, with 5 users operating on the channel. As can be seen, the performance of the blind detector approximates that of the MMSE receiver in both cases, confirming the statement that the blind detector approximates the MMSE detector. It can also be seen that the MMSE and blind detectors are converging with the decorrelating detector at higher SNRs. There is a discrepancy between the plots in Figure 3.7 due to the fact that simulation only uses 10^6 random bits (250 independent simulations with a frame length of 4000 bits) – so there are inaccuracies, especially at higher SNRs and low BERs. The theoretical plots also rely on estimated correlations, which may introduce inaccuracies. Hence the simulated plot in Figure 3.7(b) will have a lower BER performance than the theoretical plot in Figure 3.7(a). The plots in Figure 3.7 exhibit the same characteristics as those for perfect power control from [21]. From here on in this dissertation, the SNR in the figures can be assumed to be the ratio SNR = E_b/N_0.

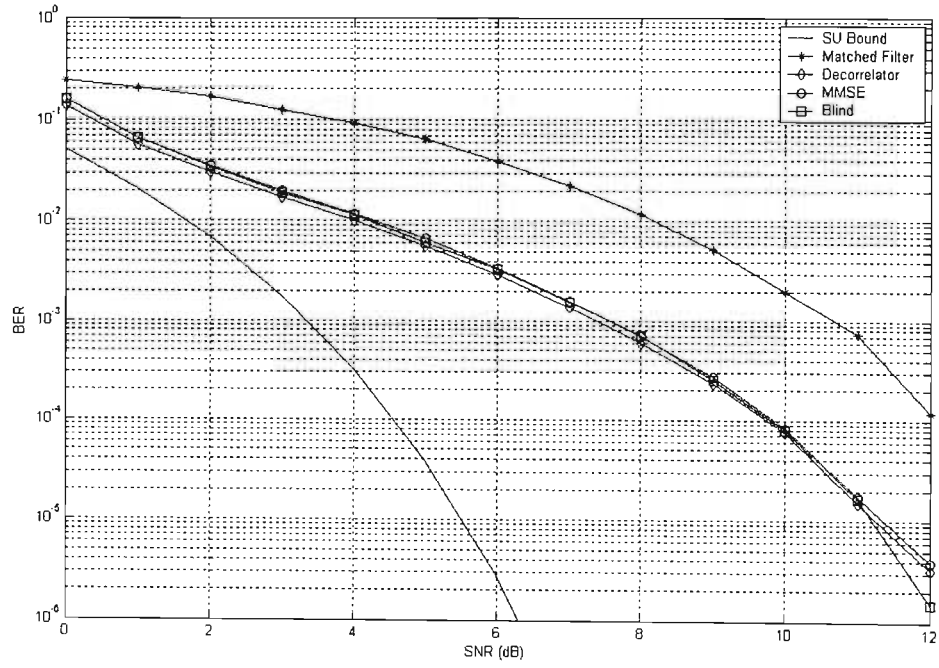
Another measure of performance that has been used is the ratio of SNRs with and without interference from other users. Equation 3.24 gives the probability of error for the kth user without interference from other users, $\gamma_k = E_k/N_0$. With multiuser interference, the user transmitting a signal with energy E_k will have an error probability P_k that exceeds $P_k(\gamma_k)$ [20]. The effective SNR γ_{ke} is the SNR with the presence of interference from other users. The efficiency is the ratio of the effective SNR, γ_k, to the SNR without multiuser interference, γ_{ke},

and it represents the loss in performance due to the multiuser interference. The measure of this performance, the asymptotic efficiency, is defined as [20]:

$$\eta_k = \lim_{N_0 \to 0} \frac{\gamma_{ke}}{\gamma_k} \tag{3.34}$$



(a)



(b)

Figure 3.7: Performance of linear MUDs for k=5 users (a) Theoretical performance (b) Simulated performance

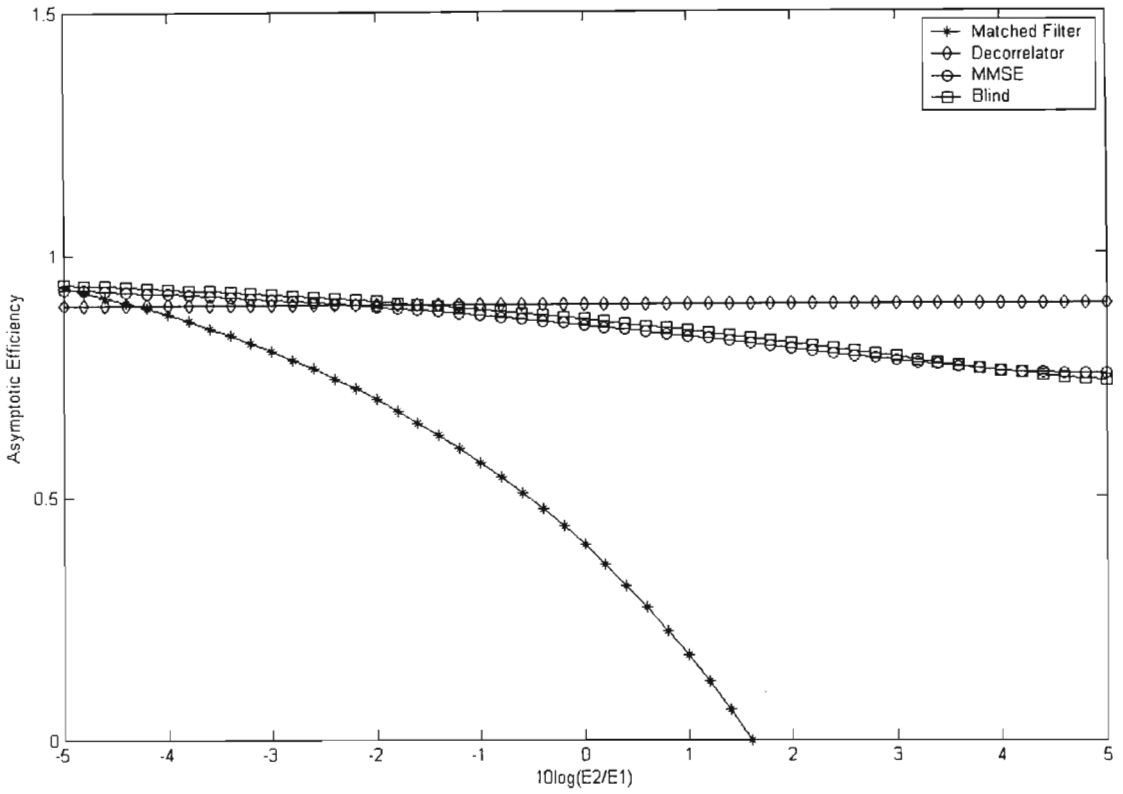


Figure 3.8: Asymptotic efficiency of various MUDs for k=2 users

Figure 3.8 shows the asymptotic efficiencies for various detectors, with two users operating on the channel. As can be seen, the conventional matched filter receiver's efficiency degrades rapidly. The MMSE and blind detectors' efficiencies are very close, and degrade, but not as rapidly as the matched filter and usually reach a steady state value. This again confirms that the blind detector approximates the performance of the MMSE detector. The decorrelating receiver, as it rejects all multiuser interference, has a constant efficiency, which is less than unity due to the noise enhancement characteristics. The characteristics shown in this plot are the same as presented in [20].

3.9 Interference Cancellation Detectors

This class of detector generates separate estimates of the MAI contributed by each user, enabling some or all of the MAI seen by any user to be subtracted out. The detectors consist of multiple stages, the decisions improving at the output of each stage.

3.9.1 Successive Interference Cancellation

Successive interference cancellation (SIC) is the serial implementation of the IC detectors. Each stage decisions, regenerates and cancels out one additional user from the received signal, reducing the MAI seen by the remaining users in the next stage. Before the first stage, the signals are ranked according to strength. The strongest signals are cancelled out first, as they

introduce more MAI than the other users. It is also easier to acquire and demodulate the strongest signals. The strongest users thus will not benefit from any reduction in MAI, whereas the weaker users could potentially see significantly reduced MAI. Each stage outputs a data decision on the strongest user present in the signal, and a modified received signal without the MAI generated by that user. The estimate for the user is generated using the data estimate, knowledge of its PN sequence and estimates of its timing, amplitude and phase [1, 6-7, 11, 13]. For users ranked 1 to K according to their signal strength, the detection rule is [13]:

$$\hat{d}_k = \mathrm{sgn}\left(y_k - \sum_{j=k+1}^{K} A_j g_j \rho_{jk} \hat{d}_j \right), \tag{3.35}$$

The SIC detector has a few implementation problems; each stage results in a longer delay and the signals need to be reordered when there is a change in the power profile. A trade-off needs to be made between the acceptable delay and the number of users that are cancelled, and another between precision of the ranking of signal powers and computational complexity. Another problem is that if the initial data estimates are incorrect, the effect on the SNR for that bit is quadrupled, thus the conventional detector needs to be reliable in order for the SIC detector to yield improvements [1, 5-7, 11, 13].

3.9.2 Parallel Interference Cancellation

Parallel interference cancellation (PIC) detectors estimate and cancel out all of the MAI for each user in parallel. The initial bit estimates from the matched filter (a decorrelating or MMSE detector may also be used) are scaled by the amplitude estimates and respread by the codes, producing a delayed estimate of the received signal, $\hat{r}_k(t - T_b)$. A partial summer then sums up all but one input signal at each output, creating the complete MAI estimate for each user [1, 5-6,13]. The result after subtracting the MAI estimates for the *k*th user, assuming perfect amplitude and delay estimation, is [13]:

$$\begin{aligned} r(t-T_b) - \sum_{i\neq k}^{K} s_i(t-Tb) &= d_k(t-\tau_k-T_b)A_k(t-\tau_k-T_b)g_k(t-\tau_k-T_b) + n(t-T_b) \\ &+ \sum_{i\neq k}^{K} \left(d_i(t-\tau_k-T_b) - \hat{d}_i(t-\tau_k-T_b) \right) A_i(t-\tau_k-T_b)g_i(t-\tau_k-T_b) \end{aligned} \tag{3.36}$$

where g(t) = ±1. This result is then passed onto a second matched filter bank to produce improved data estimates. This process can be repeated for multiple stages, each stage taking in the data estimates from the previous stage and producing a new set of estimates [1, 5-6, 13]. The soft output of stage m+1 of a PIC detector, for all bits of all users, can be represented as [13]:

$$\begin{aligned}\hat{d}[m+1] &= \mathbf{y} - \mathbf{QA}\hat{\mathbf{d}}[m] \\ &= \mathbf{Ad} + \mathbf{QA}(\mathbf{d} - \hat{\mathbf{d}}[m]) + z\end{aligned}, \tag{3.37}$$

where $\mathbf{QA}\hat{\mathbf{d}}[m]$ is an estimate for the MAI. For BPSK, the hard decision is made by taking the sign of the data estimates. Perfect amplitude and delay estimation will result in the complete cancellation of MAI [1, 5-7, 13].

A number of variations of the PIC detector exist. Using a MMSE or decorrelating detector for the initial data estimates has already been mentioned. As the IC detectors are heavily reliant on the initial data estimates, using one of these detectors will reduce the number of initial estimates in error, resulting in better performance. Bits that have already been detected at the output of the current stage can be used to improve the detection of the remaining bits in the same stage. Linearly combining the soft-outputs of different stages capitalises on noise correlation and causes cancellation amongst noise terms [13].

3.10 Conclusion

The need for multiuser detection was discussed in this chapter, and a number of multiuser detection schemes were described. The chapter focussed on the blind MOE method, as this forms an integral part of the proposed detector. The Equations for the blind MOE detector were derived. Three multiuser detectors (the MMSE, blind and decorrelating detectors) were simulated and their performances were compared. An overview of IC detectors was given as they are used in iterative multiuser receivers described in Section 5.1.3.

CHAPTER 4 ERROR CODING

This chapter will look at various error coding schemes, focusing on convolutional and turbo codes, components of which are needed for iterative multiuser detection. The decoding techniques for convolutional codes and turbo codes are derived. Convolutional codes and turbo codes were simulated, and the performances of the iterative decoding techniques were compared. These error-correcting codes are of particular interest as the iterative MUD proposed in chapter 5 is based on a concatenated structure similar to that of turbo-codes, and are commonly used for FEC in CDMA based 3G communication systems. The iterative MUDs appearing in literature also use convolution codes and turbo-codes as part of their structure.

4.1 Block Codes

4.1.1 General Block Code Structure and Performance

For each block of k information bits, r redundant bits are added, hence each message is now a codeword of n bits with [3]:

$$n = k + r. \tag{4.1}$$

Codes that have been created by taking a block of k information bits and introducing $r = n - k$ redundant bits to form a codeword are called block codes and are designated (n,k) codes. Codewords of length n, consisting of k information bits and r redundant bits are systematic, and have a rate [3]:

$$R_c \equiv k/n. \tag{4.2}$$

If it is required for the n bit codeword to be transmitted in the same amount of time as the original k bit message, then $nT_c = kT_b$, where T_c and T_b are the code and bit intervals respectively [3]. Given that $f_b = 1/T_b$ and $f_c = 1/T_c$, then

$$\frac{f_c}{f_b} = \frac{T_b}{T_c} = \frac{n}{k} = \frac{1}{R_c}. \tag{4.3}$$

The distance between two codes is defined as the number of bits that differ in two codes. The Hamming distance d_{min} of a code is then the minimum distance between two or more codewords. As it is most likely that two codewords will be confused when the distance of the code is a minimum, then the Hamming distance provides an upper bound to the effectiveness of a code [3]. Assuming there are D errors in a received codeword, it will be possible to detect with certainty that the received codeword is not valid, provided

$$D \leq d_{\min} - 1. \tag{4.4}$$

For t errors in the received codeword, the errors can be corrected provided

$$2t + 1 \leq d_{\min} \text{ (odd), or} \tag{4.5a}$$

$$2t + 2 \leq d_{\min} \text{ (even).} \tag{4.5b}$$

Hence for $d_{min} = 9$ an invalid code will be detected if there are 8 or less bits in error, and these errors will be able to be corrected if there are 4 or less bits in error [3].

4.1.2 Coding and Decoding of Block Codes

The code $\mathbf{C} = [c_1 \quad c_2 \quad \cdots \quad c_n]$ for a message block $\mathbf{A} = [a_1 \quad a_2 \quad \cdots \quad a_k]$ is generated by

$$\mathbf{C} = \mathbf{AG}, \tag{4.6}$$

where **G** is the generator matrix and is given by

$$\mathbf{G} = \left[\mathbf{I}_k | \mathbf{P}\right]_{kxn}, \tag{4.7}$$

where **P** generates the check bits at the end of the codeword, and $\mathbf{I}_k$ is the kxk identity matrix. Selection of the matrix **P** defines properties of the code such as the number of errors that can be corrected and the ability to correct random and burst errors. Associated with each (n,k) block code is a parity check matrix [3]:

$$\mathbf{H} = \left[\mathbf{P}^{\mathrm{T}} | \mathbf{I}_{n-k}\right]_{(n-k)xn}, \tag{4.8}$$

which is used to verify whether a codeword is generated by the matrix **G**, such that

$$\mathbf{GH}^{\mathrm{T}} = \mathbf{0}. \tag{4.9}$$

If the codeword **C** is transmitted, but a vector $\mathbf{R} = \mathbf{C} \oplus \mathbf{E}$ is received, where **E** is the error word and $\oplus$ denotes modulo-2 addition, the receiver computes the syndrome as [3]:

$$\mathbf{S} = \mathbf{HR}^{\mathrm{T}} \tag{4.10}$$

$$= \mathbf{H}(\mathbf{C} \oplus \mathbf{E})^{\mathrm{T}}$$

$$= \mathbf{HC}^{\mathrm{T}} \oplus \mathbf{HE}^{\mathrm{T}}$$

$$= \mathbf{HE}^{\mathrm{T}}, \tag{4.11}$$

which is non-zero. The syndrome is used to detect errors, and can also be used to correct them. The syndrome **S** for an error in the *j*th element represents the *j*th column of the **H** matrix. For multiple errors, the syndrome is the sum of those columns of the **H** matrix corresponding to the error locations [3]:

$$S = \sum_i e_i h_i^{\mathrm{T}} , \tag{4.12}$$

where h_i^{T} is the *i*th row of $\mathbf{H}^{\mathrm{T}}$ matrix and e_i is the *i*th element of the error word **E**.

4.1.3 Examples of Codes

Single parity-check bit codes can only detect one error, but cannot correct it. The check bit is selected such that

$$d_1 \oplus d_2 \oplus \cdots \oplus d_k \oplus c_{k+1} = 0 , \tag{4.13}$$

and has r=1 and n=k+1.

In repeated codes, a bit is represented by $2t$+1 of that bit i.e. a binary '0' is represented by $2t$+1 '0's and a binary '1' is represented by $2t$+1 '1's. Repeated codes are block codes as the redundant bits are determined by Equation (4.8). The (n = $2t$ + 1, k = 1) repeated code can correct t errors, but is inefficient as it has a rate of 1/($2t$+1), thus it requires a large bandwidth [3].

The rows of the Hadamard matrix form the codeword in the Hadamard code. The Hadamard matrix is a square (n x n) matrix with n =$2k$. One codeword consists of all zeroes and all the other codewords have n/2 '0's and n/2 '1's, each codeword differing from the others in n/2 places. As a result, the codewords are orthogonal. The Hadamard matrix that provides two codewords is

$$\mathbf{M}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.14}$$

The matrix that provides four codewords can be given by

$$\mathbf{M}_4 = \begin{bmatrix} \mathbf{M}_2 & \mathbf{M}_2 \\ \mathbf{M}_2 & \mathbf{M}_2^* \end{bmatrix}$$

where $\mathbf{M}_2^*$ is the $\mathbf{M}_2$ matrix with each element replaced by its complement. Generally it can be written [3]:

$$\mathbf{M}_{2n} = \begin{bmatrix} \mathbf{M}_n & \mathbf{M}_n \\ \mathbf{M}_n & \mathbf{M}_n^* \end{bmatrix} \tag{4.15}$$

In the n bit codeword there are $r = n - k = 2^k - k$ parity bits. As k increases the number of parity bits will increase significantly, becoming considerably large in comparison with the number of k information bits, resulting in a small rate

$$R_c = \frac{k}{n} = \frac{k}{2^k} = k2^{-k}. \tag{4.16}$$

As the Hadamard code is orthogonal, the Hamming distance is

$$d_{\min} = \frac{n}{2} = 2^{k-1}. \tag{4.17}$$

and the number of errors that can be corrected is [3]:

$$t = \frac{n}{4} - 1 = 2^{k-2} - 1, \tag{4.18}$$

hence it is required that $k > 2$ for error correction. Due to t increasing by 2^k for large k, significant error correction is possible.

The Bose, Chaudhuri, and Hocquenghem (BCH) codes employ k information bits and r parity check bits, resulting in a codeword of length $n = k + r$. The number of errors that can be corrected is [3]:

$$t = \frac{r}{m}, \tag{4.19}$$

where m is an integer related to the number of bits in the codeword by

$$m = \log_2(n+1). \tag{4.20}$$

The minimum distance between the BCH codes is related to the number of correctable errors, t, by the inequalities given in Equation (4.5). BCH codes form a large class of error correcting codes to which Hamming codes, to be discussed below, and Reed-Solomon codes, to be discussed in Section 4.2, belong.

The Hamming code is one that has a minimum distance of d_{min}=3, hence only a single error can be corrected (t=1). The n bits of the codeword, the k bits of the uncoded word, and the number of parity bits (r) are related by [3]:

$$n = 2^r - 1, \tag{4.21}$$

$$k = 2^r - 1 - r. \tag{4.22}$$

The parity check matrix has r rows and n columns. Each column is unique and consists of r elements, and no column consists of all zeroes. There are $2^{n-k} - 1$ distinct columns excluding the zeroes column. The last r columns of **H** must form the identity matrix $\mathbf{I}_r$.

4.2 Burst Error Correction

Whilst the average bit error rate may be small, the error correcting codes discussed above may not be sufficient to correct the errors that are clustered. In other words, for a received bit stream there may be a large percentage of errors in one section, yet the rest of the bit stream remains relatively error free. When errors are clustered they are said to have occurred in bursts.

4.2.1 Block Interleaving

For a transmission of kl data bits, the bits are loaded into a shift register organised into k rows of l consecutive bits. The bit enters at position d_{11}, and for each shift it moves one position to the right. When the bit at the end of a row is shifted, it moves to the first position of the next row (d_{1l} to d_{21}), as shown if Figure 4.1. When the shift register is full the data stream is diverted to a second similar shift register where coding is applied to the data stored in the first shift register. The bits in a column are viewed as an uncoded word for which parity checks are generated, resulting in the 'codeword' $a_{11}\, a_{21} \cdots a_{k1}\, c_{11}\, c_{21} \cdots c_{r1}$ consisting of k information bits and r parity bits. When the coding is complete the information is transmitted, row-by-row, in the order $c_{rl} \cdots c_{r1} \cdots c_{1l} \cdots c_{11}\, a_{kl} \cdots a_{k1} \cdots a_{2l} \cdots a_{21}\, a_{1l} \cdots a_{12}\, a_{11}$, which is the same order that it originally entered the register, only now the parity bits are transmitted [3].

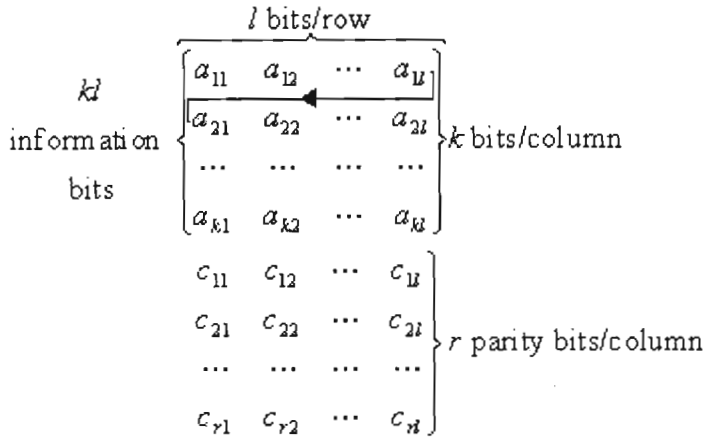


Figure 4.1: Block interleaver registers

The received data is stored in the same order as in the transmitter, and error correction decoding is performed, after which the parity bits are discarded and the data shifted out of the register. Each column can correct one error, and assuming there is a burst of l errors (in consecutive bits), each column will have one error, and thus all errors can be corrected. If there is a burst of more than l errors, then error correction for the columns with more than one error is not assured. If the code is able to correct t errors, then the interleaving will allow for the correction of a burst of B errors with [3]:

$$B \leq tl. \tag{4.23}$$

4.2.2 Convolutional Interleaving

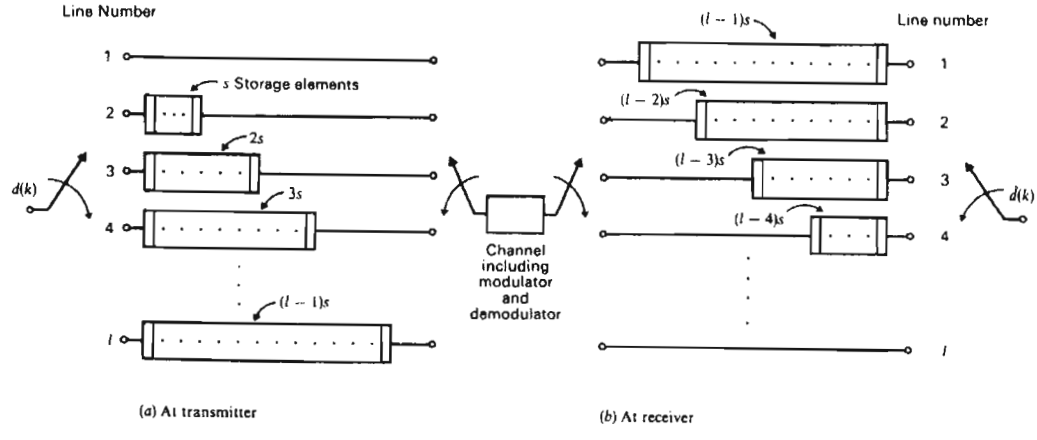


Figure 4.2: Convolutional interleaving [3]

Convolutional interleaving is shown in Figure 4.2. The four switches are synchronised, making contact with each line at the same time, before moving on to the next line, returning to line 1 after line l. The switches move from line to line at the bit rate of the input $d(k)$. Each line has shift registers as storage elements. On the transmitter side, line 1 has no storage elements, and

the number of elements increases by s at each consecutive line, line l have $(l-1)s$ storage elements. At the receiver side the storage elements go in the reverse order, line 1 has $(l-1)s$ elements, and line l has none, consequently each line has the same number of storage elements, being $(l-1)s$ [3].

During a particular bit interval there is contact at both the input and output of line l_i. At the end of the bit interval a clock signal causes the shift register of only line l_i to shift the bit at the input into the leftmost storage elements, and the bits in the storage elements one position to the right. The switch moves onto the next line, l_{i+1} when the process has started, so due to transmission delay of the shift register, the new bit at the output of line l_i will only be noticed until the next time the switch makes contact with the line. It is to be noted that the clocks that drive the switches have a rate f_b, which is the bit rate, whereas the clocks that drive the shift registers have a rate of f_b/l. The shift registers are activated sequentially, just as the switch is about to break contact with its line [3].

Now consider that the all the shift registers, in both the transmitter and receiver, are initially short circuited. If bit $d(k)$ appears on line l_i, then the corresponding bit $\hat{d}(k)$ appears at the output immediately. The next input bit, $d(k+1)$ will be the next received bit, except that it will be on line l_{i+1}. In other words, the received sequence will be the same as the transmitted sequence. With the shift registers in place, each of the lines will have a delay of $(l-1)s$, therefore the output sequence will still be the same as the input sequence, except that $\hat{d}(k)$ will be delayed by $(l-1)s$ with respect to $d(k)$. The sequence over the channel, however, is different in that it is interleaved. For adjacent bits $d(k)$ and $d(k+1)$, $d(k)$ will still be in the same position, but the next bit will become $d(k+1+ls)$ i.e. there will be ls bits in between two previously consecutive bits. The convolutional interleaver requires less memory than the block interleaver for the same interleaving distance, and the interleaving structure can be easily changed by altering the number of lines l, or the increment in the number of storage elements per line, s [3].

4.2.3 Reed-Solomon Codes

The block codes described above are organised on the basis of bits, whereas the Reed-Solomon (RS) codes are organised on the basis of symbols. Dealing with symbols, however, means that if just one bit is in error, then that entire symbol is in error. The RS code has k information symbols, r parity symbols and a codeword length of $n = k + r$ symbols. The number of symbols in a codeword is [3]:

$$n = 2^m - 1, \tag{4.24}$$

and is able to correct errors in t symbols, where

$$t = r/2. \tag{4.25}$$

The RS codes are not efficient at correcting random errors, but more suited to burst errors. For example a (255,233) RS code can correct 128 consecutive bit errors if it has an error-free region of 239 symbols (1912 bits). If the errors are random, and there is at most one error per symbol, then the RS code can only correct sixteen bit errors in 2040 bits [3].

4.3 Convolutional Codes

4.3.1 Encoding

In convolutional coding, the parity check bits are calculated over a span of information bits. Bits are stored in a storage device, such as a flip-flop or a shift register. The number of bits stored is the constraint span, k. The flip-flops (or the elements of the shift register) are connected to v modulo-2 adders (EXCLUSIVE–OR logic summers), and a commutater samples the outputs of the summers. For each new bit shifted into the first storage element, the summer outputs are sampled by the commuter, thus each bit effects the output for the number of storage elements. The number of bits at the output is given by $v(L+k)$, where L is the number of input bits, and is generally much larger than k, so the number of bits at the output can be approximated as $v(L+k) \approx vL$ [3].

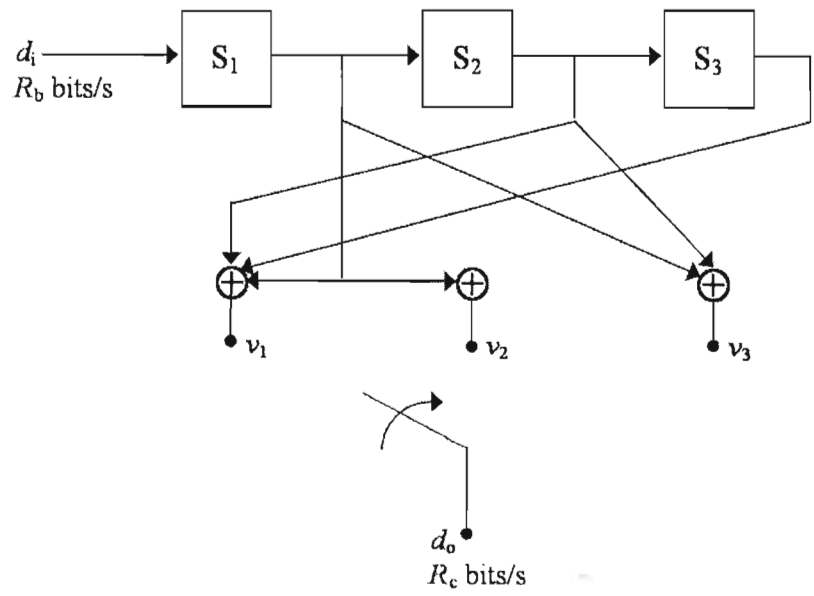


Figure 4.3: An example of a convolutional encoder

In Figure 4.3, the constraint span is k = 3, with v = 3 summers, and each input message bit produces the sequence $v1$ $v2$ $v3$, which are calculated by [3]:

$$v_1 = s_1 \oplus s_2 \oplus s_3$$
$$v_2 = s_1$$
$$v_3 = s_1 \oplus s_2$$

Thus, for an input sequence, d_i, of 1 0 1 0 0 1, the resulting output sequence, d_o, will be 111 101 011 101 100 111. As can be seen, for every input bit, there are three output bits, hence the code rate R_c=1/3. In general, the rate is given by

$$Rc=1/v. \tag{4.26}$$

4.3.2 Decoding

The code tree is one of the searching techniques by which convolutional codes may be decoded. The code tree is searched for the path closest in Hamming distance to the received sequence. Figure 4.4 on the next page is an example of a code tree. Generally the starting point is at the left, and corresponds to the situation before the occurrence of the first message bit. Another convention is that a downwards shift in the path signifies a '1', whilst an upwards direction denotes a '0'. Storage of the code tree is impractical for large constraint spans, as there are 2^k branches.

If the sequence 1 0 1 0 is fed into the encoder, the output of the encoder (and input of the decoder, assuming no errors) will be 111 101 011 101. From the start point, there are only two options; $v_1v_2v_3 = 000$, and $v_1v_2v_3 = 111$. The path taken is to 111, hence the bit is a '1'. The next three bits in the received sequence is 101, taken the path up, denoting a '0'. The following sets of three bits are 011 and 101, taking the path down then up, signifying a '1' and '0' respectively. The solution is then 1010. If there is an error in the first set of bits, and they are received as 101, then the path taken will still be down, as 101 compared to 111 has only one different bit, but there are two different bits when compared to 000.

Trellis diagrams are a compact form of the code tree: it avoids redundancy but still provides an effective representation of the convolutional coder's response to the input bit stream. If the contents of the shift register prior to the inputting of the current information bit determine the state, then the state transitions shown in Table 4.1 can be realised. These transitions are for the example given above. Figure 4.5 shows the state diagram for the transitions given in Table 4.1. In the figure, the top value for the transition is the new input bit, and the bottom value, in brackets, is the values for $v_1v_2v_3$.

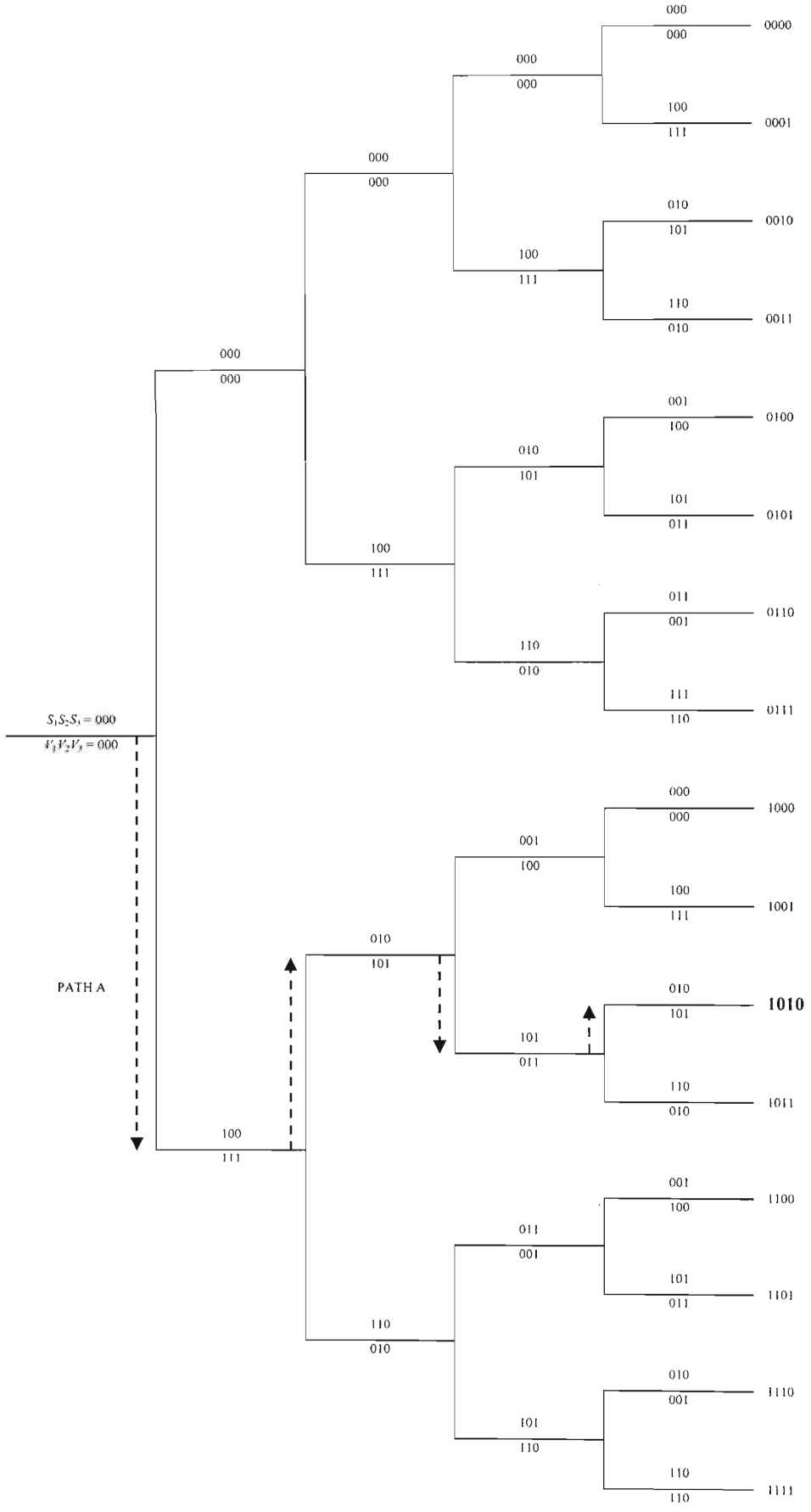


Figure 4.4: Code Tree

Table 4.1: State transition example

Previous			Input	Current				Output
State	S1	S2		S1	S2	S3	State	
A	0	0	0	0	0	0	A	000
			1	1	0	0	C	111
B	0	1	0	0	0	1	A	100
			1	1	0	1	C	011
C	1	0	0	0	1	0	B	101
			1	1	1	0	D	010
D	1	1	0	0	1	1	B	001
			1	1	1	1	D	110

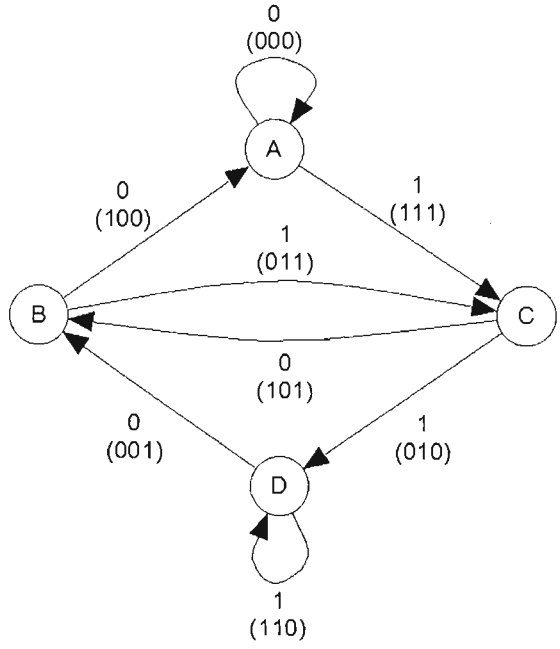


Figure 4.5: State diagram

Figure 4.6 shows the trellis diagram for the first three clock intervals for the example above. The values in brackets alongside the nodes are the numbers of received bits in error. The path with the least (or no) errors is the 'correct' path. The triple-digit value is the output, $v_1v_2v_3$. From this the input bit can be calculated.

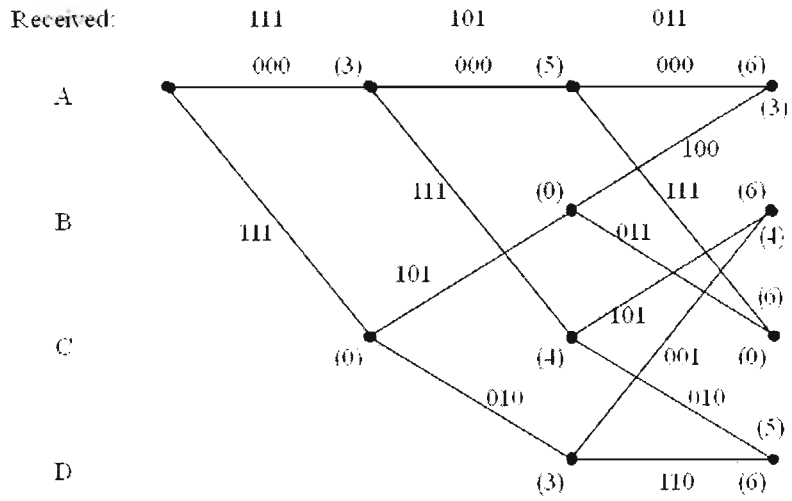


Figure 4.6: Trellis diagram

The Viterbi algorithm is an efficient algorithm for searching the trellis diagram for the most likely path. It relies on the fact that paths entering a node in a trellis will be identical onward from that point. For Viterbi decoding, the metric for the single path entering each state of the encoder is computed. If multiple paths connect to the same node, discard the one with the largest cumulative discrepancy (the errors in Figure 4.6). The surviving paths are stored, and this process is continued until the end of the message. As this algorithm discards the path with the most cumulative errors, the amount of information that is stored is reduced. Figure 4.7 shows the survivors for the trellis in Figure 4.6.

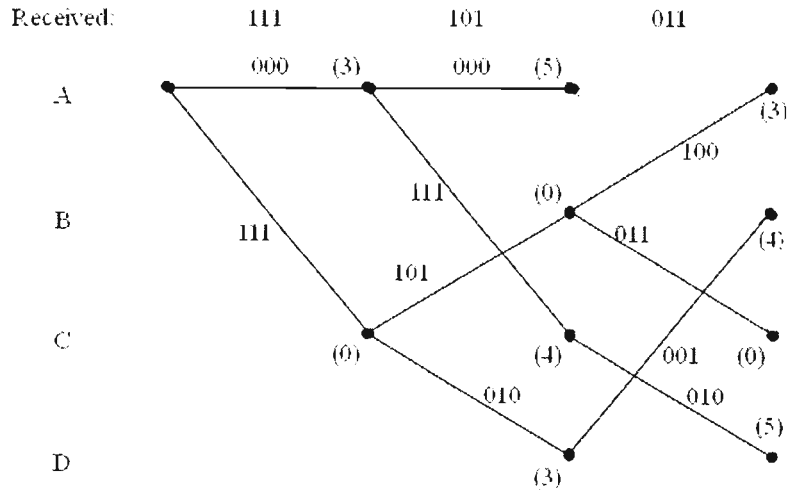


Figure 4.7: Survivors

4.4 Turbo Codes

Turbo codes have received much attention due to its near Shannon limit performance, and may become the premier error control code [22]. Whilst there are terrestrial communication systems that can perform at near Shannon limit, Turbo codes can match this performance for satellite and deep-space communications [23-25]. Their performance is due to two concepts: that of concatenation and iterative decoding. The sections below give an overview of the encoding and decoding methods for a turbo encoder.

4.4.1 Encoder

Benedetto et al described Turbo codes as being "parallel concatenated convolutional code employing two (normally equal) rate-k/n systematic convolutional encoders and an interleaver" [25]. The (n – k) check sequences of the first encoder are transmitted together with the k information sequence. The same k information sequences are interleaved and enter the second encoder; the (n – k) check sequences generated by the second encoder are also transmitted. The rate of the code is then k/(2n – k) [25].

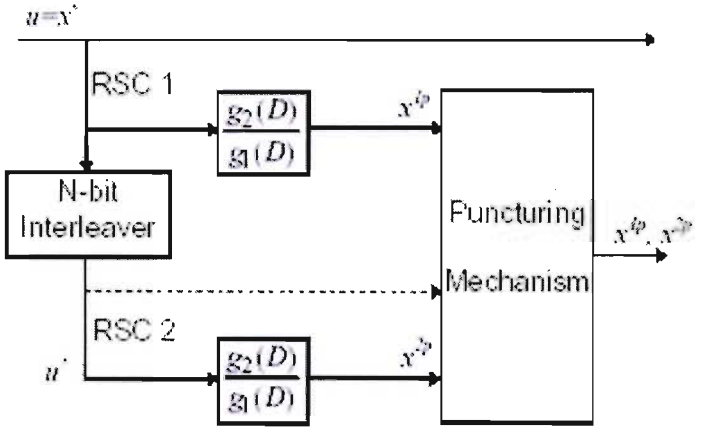


Figure 4.8: Turbo encoder [23]

Figure 4.8 shows an example of an encoder, with a generator matrix of:

$$\left[1 \quad \frac{g_2(D)}{g_1(D)}\right].$$

The puncturing mechanism is optional. Its function is to periodically delete bits to reduce coding overhead. For its application in turbo codes it is preferable to only delete parity bits [23]. Without it, the resulting code is of rate 1/3, but will exhibit improved performance over the rate 1/2 code as a result of puncturing. Woodard and Hanzo [24] discuss various puncturing methods

and various interleaver designs and conclude that a simple and effective method to use for the interleaver is "odd-even separation", where the odd and even bits are kept separate. Using this with alternate puncturing of the parity bits of the component codes (the most common puncturing method), it can be shown that one and only one of the parity bits associated with each information bit will be left unpunctured. In other cases none of the parity bits were transmitted, or parity bits from both component codes were transmitted.

The block interleaver is used to distribute possible errors amongst different codewords, as the inner decoder may be overwhelmed due to error propagation, and be unable to correct the errors if there are too many in a single codeword [22]. The functioning of the block-interleaver was dealt with in Section 4.2.1. For is application to turbo codes, Ryan, suggests that the interleaver be made large, and assumes that $n \geq 1000$ in [23].

A class of infinite impulse response (IIR) convolutional codes, also called recursive systematic convolutional (RSC) encoders due to the fact that they feed back previously decoded bits back to the encoder's input, have been shown to be essential for the high performance of turbo-codes [23-26,28-29]. For high code rates they exhibit better performance than non-systematic convolutional codes (NSC), for any SNR [26]. An example of an RSC encoder is shown in Figure 4.9.

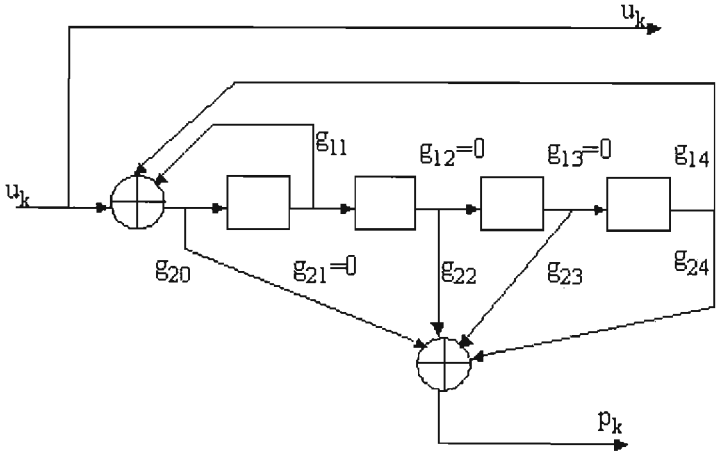


Figure 4.9: Example of a RSC encoder (g_1, g_2) = (31, 27) [23]

4.4.2 Decoding

A ML decoder is not suitable for turbo codes: for a codeword length $n \geq 1000$, the ML decoder would have to compare 2^n code sequences to the noisy received sequence in order to choose the codeword with the best correlation. Viterbi's algorithm, which allows elimination of possible codewords, is also not suitable as the presence of the interleaver complicates the trellis structure

of the turbo codes, making the codes look more like block codes [23]. Prior to the advent of turbo codes, decoding strategies for concatenated codes were receiving much attention. These decoding strategies used multiple decoders that operated cooperatively and iteratively. Two types of decoders were receiving interest: a soft-output Viterbi algorithm (SOVA), and a symbol-by-symbol maximum *a posteriori* (MAP) algorithm [23]. These two algorithms will be discussed in more detail below.

In the MAP decoder, which operates on a symbol-by-symbol basis, the decoder decides $u_k = +1$ if $P(u_k = +1|\mathrm{y}) > P(u_k = -1|\mathrm{y})$, and decides $u_k = -1$ otherwise, where y is the received signal. In other words, the decision $\hat{u}_k$ is given by $\hat{u}_k = \mathrm{sign}[L(\hat{u}_k)]$, where

$$L(\hat{u}_k) \equiv \log\left(\frac{P(u_k = +1 \mid \mathrm{y})}{P(u_k = -1 \mid \mathrm{y})}\right), \tag{4.27}$$

which is known as the log *a posteriori* probability (LAPP) ratio. Taking into account the code's trellis, this may be rewritten as:

$$L(u_k) = \log\left(\frac{\sum_{S+} p(s_{k-1} = s', s_k = s, y)/p(\mathrm{y})}{\sum_{S-} p(s_{k-1} = s', s_k = s, y)/p(\mathrm{y})}\right), \tag{4.28}$$

where $s_k \in S$ is the state of the encoder at time k, $S+$ is the set of ordered pairs (s',s) corresponding to all state transitions $(s_{k-1} = s') \rightarrow (s_k = s)$ due to the data input $u_k = +1$, and similarly $S-$ is the set of ordered pairs for the data input $u_k = -1$ [23, 24]. In (4.28) $p(\mathbf{y})$ may be cancelled by applying Bayes' Rule, meaning only an algorithm for computing $p(s', s, y) = p(s_{k-1} = s', s_k = s, y)$ is required:

$$p(s', s, \mathrm{y}) = \alpha_{k-1}(s')\gamma_k(s', s)\beta_k(s), \tag{4.29}$$

where $\gamma_k(s',s) \equiv p(s_k = s, y_k \mid s_{k-1} = s')$, $\alpha_k(s) \equiv p(s_k = s, \mathrm{y}_1^k)$ is calculated recursively as $\alpha_k(s) = \sum_{s' \in S} \alpha_{k-1}(s')\gamma_k(s',s)$ with initial conditions $\alpha_0(0) = 1$ and $\alpha_0(s \neq 0) = 0$. $\beta_k(s) \equiv p(\mathrm{y}_{k+1}^N \mid s_k = s)$ is calculated in a backward recursion $\beta_{k-1}(s') = \sum_{s \in S} \beta_k(s)\gamma_k(s',s)$ and has boundary conditions $\beta_N(0) = 1$ and $\beta_N(s \neq 0) = 0$ [23,26-28].

If the divisor p(**y**) in (4.28) is cancelled out the algorithm becomes unstable [23] an algorithm with modified probabilities is required. Dividing (4.28) by $p(\mathbf{y})/p(y_k)$ the following algorithm with modified probabilities $\tilde{\alpha}_k$ and $\tilde{\beta}_k$ is obtained:

$$p(s', s \mid \mathrm{y}) p(y_k) = \tilde{\alpha}_{k-1}(s') \gamma_k(s', s) \tilde{\beta}_k(s), \tag{4.30}$$

where

$$\tilde{\alpha}_k(s) = \frac{\sum_{s'} \alpha_{k-1}(s') \gamma_k(s's)}{\sum_s \sum_{s'} \alpha_{k-1}(s') \gamma_k(s's)} = \frac{\sum_{s'} \tilde{\alpha}_{k-1}(s') \gamma_k(s's)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s') \gamma_k(s's)},$$

and

$$\tilde{\beta}_{k-1}(s') = \frac{\sum_s \tilde{\beta}_k(s) \gamma_k(s', s)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s') \gamma_k(s', s)}.$$

By combining (4.27) and (4.29), the modified algorithm is obtained:

$$L(u_k) = \log\left(\frac{\sum_{S+} \tilde{\alpha}_{k-1}(s') \gamma_k(s', s) \tilde{\beta}_k(s)}{\sum_{S-} \tilde{\alpha}_{k-1}(s') \gamma_k(s', s) \tilde{\beta}_k(s)} \right). \tag{4.31}$$

From Bayes' rule, the LAPP ratio can be written as:

$$L(u_k) = \log\left(\frac{P(\mathrm{y} \mid u_k = +1)}{P(\mathrm{y} \mid u_k = -1)} \right) + \log\left(\frac{P(u_k = +1)}{P(u_k = -1)} \right), \tag{4.32}$$

the second term being the *a priori* information, which is typically zero for conventional decoders as $P(u_k = +1)$ is usually equal to $P(u_k = -1)$. For iterative decoders, the first component decoder receives extrinsic information from the second component decoder for each u_k which acts as *a priori* information. Likewise, the second component decoder receives extrinsic information from the first decoder, which acts *a priori* information [23,24]. Such a decoder can be seen in Figure 4.10.

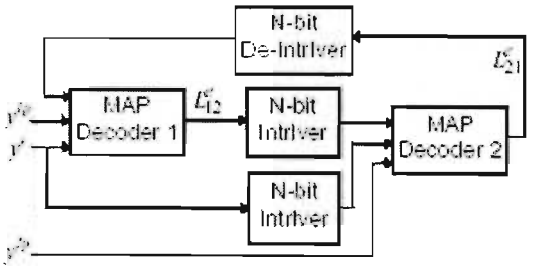


Figure 4.10: Turbo decoder employing MAP decoders [23]

Defining the *a priori* information as [23]:

$$L^e(u_k) \equiv \log\left(\frac{P(u_k = +1)}{P(u_k = -1)}\right),$$

which is the second term of (4.31), and observing that

$$\begin{aligned}\gamma_k(s',s) &\approx \exp\left[\frac{1}{2}u_k\left(L^e(u_k) + L_c y_k^s\right) + \frac{1}{2}L_c y_k^p x_k^p\right] \\ &= \exp\left[\frac{1}{2}u_k\left(L^e(u_k) + L_c y_k^s\right)\right]\gamma_k^e(s',s)\end{aligned} \tag{4.33}$$

where $L_c \equiv \frac{4E_c}{N_0}$ and $\gamma_k^e(s',s) \equiv \exp\left[\frac{1}{2}L_c y_k^p x_k^p\right]$. By combining Equations (4.31) and (4.33) the following probability is obtained [23, 24]:

$$L(u_k) = L_c y_k^s + L^e(u_k) + \log\left(\frac{\sum_{S+}\tilde{\alpha}_{k-1}\gamma_k^e(s',s)\tilde{\beta}_k(s)}{\sum_{S-}\tilde{\alpha}_{k-1}\gamma_k^e(s',s)\tilde{\beta}_k(s)}\right). \tag{4.34}$$

The first term is the channel value, the second term is the *a priori information* about u_k from a previous decoder, and the third term represents the extrinsic information that can be passed on to any subsequent decoder [23,24]. For a more in-depth derivation of the probabilities, refer to [23, 24, 26-28].

The SOVA has two main differences from the Viterbi algorithm: the path metrics are modified to take account of *a priori* information, and it provides *a posteriori* information as a soft output. To modify the Viterbi algorithm to take into account the a priori information, the state sequence $\underline{s}_k^s$, giving the states along the surviving path at state $S_k = s$ at stage k in the trellis, needs to be considered. The probability of the path being correct is [24]:

$$p(\underline{s}_k^s \mid \underline{y}_{j\le k}) = \frac{p(\underline{y}_{j\le k}, s_k^s)}{p(\underline{y}_{j\le k})}. \tag{4.35}$$

The probability of the received sequence for transitions up to and including the kth transition is constant for all paths through the trellis to stage k, the probability that the path is correct is proportional to $p(y_{j\le k}, s_k^s)$. The metric should thus be defined so that maximising the metric will maximise $p(y_{j\le k}, s_k^s)$. The metric should also be easily computable in a recursive manner going from the (k–1)th stage to the kth stage in the trellis. If the path $\underline{s}_k^s$ at the kth stage has the

path $\underline{s}_{k-1}^{s'}$ for the first k–1 transitions, then using the definition $\gamma_k(s',s) \equiv p(s_k = s, y_k \mid s_{k-1} = s')$, we get [24,30]:

$$\begin{aligned} p(\underline{y}_{j\le k}, \underline{s}_k^s) &= p(\underline{y}_{j\le k-1}, \underline{s}_{k-1}^{s'}) p(\underline{y}_k, s \mid s') \\ &= p(\underline{y}_{j\le k-1}, \underline{s}_{k-1}^{s'}) \gamma_k(s',s) \end{aligned} \tag{4.36}$$

The metric for the path $\underline{s}_k^s$ is

$$M(\underline{s}_k^s) = M(s_{k-1}^{s'}) + \frac{1}{2} u_k L(u_k) + \frac{L_c}{2} \sum_{l=1}^{n} y_{kl} x_{kl} \,. \tag{4.37}$$

The metric is now updated as in the Viterbi algorithm, but with the $u_k L(u_k)$ term to take into account the a priori information [24,30].

The following paragraph discusses the modification to the Viterbi algorithm to allow it to give soft outputs. There are two paths reaching state $S_k = s$ at stage k in a binary trellis. The modified Viterbi algorithm given by (4.37), calculates the metric for both merging paths, discarding the path with the lowest metric. Let the two paths reaching state $S_k = s$ be $\underline{s}_k^s$ and $\hat{\underline{s}}_k^s$, with metrics $M(\hat{\underline{s}}_k^s)$ and $M(\underline{s}_k^s)$, respectively. If the path $\underline{s}_k^s$ is selected as the survivor due to its having a higher metric, the metric difference Δ_k^s can be defined as [24]:

$$\Delta_k^s = M(\underline{s}_k^s) - M(\hat{\underline{s}}_k^s) \ge 0. \tag{4.38}$$

The probability that the decision to select $\underline{s}_k^s$ and discard $\hat{\underline{s}}_k^s$ is correct is given by:

$$\begin{aligned} P(\text{correct decision at } Sk = s) &= \frac{P(\underline{s}_k^s)}{P(\underline{s}_k^s) + P(\hat{\underline{s}}_k^s)} \\ &= \frac{e^{M(\underline{s}_k^s)}}{e^{M(\underline{s}_k^s)} + e^{M(\hat{\underline{s}}_k^s)}} \\ &= \frac{e^{\Delta_k^s}}{1 + e^{\Delta_k^s}} \end{aligned} \tag{4.39}$$

and the LLR that this is the correct decision is simply given by Δ_k^s. When the end of the trellis has been reached and the ML path through the trellis has been identified, the LLRs giving the reliability of the bit decisions along the path need to be found. In the Viterbi algorithm, all

surviving paths at stage l, in the trellis will normally have come from the same path at some point before l in the trellis, the point is taken to be at most δ transitions before l, δ usually being set to five times the constraint length of the convolutional code [24]. The value of a bit may therefore have been different if the Viterbi algorithm had selected one of the paths that merged with the ML path up to δ transitions later, instead of the ML path. If the algorithm selected any of the paths that merged with the ML path after this point, the value of the bit would not be affected. Thus, when calculating the LLR of a bit, SOVA must account for the probability that the paths merging with the ML path from stage k to $k + \delta$ in the trellis were incorrectly discarded. This is done by considering the metric difference $\Delta_i^{s_i}$ for all states s_i along the ML path from stage $i = k$ to $i = k + \delta$ [24]. It is shown in [31] that this LLP can be approximated by

$$L(u_k \mid \underline{y}) \approx u_k \min_{\substack{i=k\ldots k+\delta \\ u_k \neq u_k^i}} \Delta_i^{s_i}, \tag{4.40}$$

where u_k^i is the value of the bit for the path which merged with the ML path that was discarded at stage i. The minimization is only carried out for the paths merging with the ML path which would have resulted in a different value for the bit u_k if it had been selected as a survivor. Paths that merge with the ML path, but do not change the value of u_k, do not affect the reliability. For a more in-depth derivation of the probabilities, refer to [24, 30,31].

4.4.3 Effects on Turbo Code Performance

Figure 4.11 gives a comparison of the performance of the two decoding algorithms with and without puncturing over an AWGN channel in order to illustrate the effects of the decoding algorithm and puncturing on performance. The generator matrix for the simulated turbo codes is $\mathbf{g}$ = [1 1 1, 1 0 1], with eight iterations being performed to reach the final decision on the bit value.

As can be seen from Figure 4.11, the log-MAP algorithm performs better than the SOVA. This is supported by Pietrobon in [32] and the results generated by Woodard and Hanzo in [24]. Both papers state that the MAP algorithms can outperform the SOVA by 0.5 dB or more, especially at low values of SNR or high values of BER [32]. The figure also illustrates that un-punctured turbo codes (rate-1/3) perform better than the punctured (rate-1/2) codes. At a BER of 10^{-4} a gain of approximately 0.9 dB can be seen for the MAP algorithm, and a gain of 0.4 dB for the SOVA. Similar results are given in [24], where it is also stated that the gain can be up to 2.4 dB.

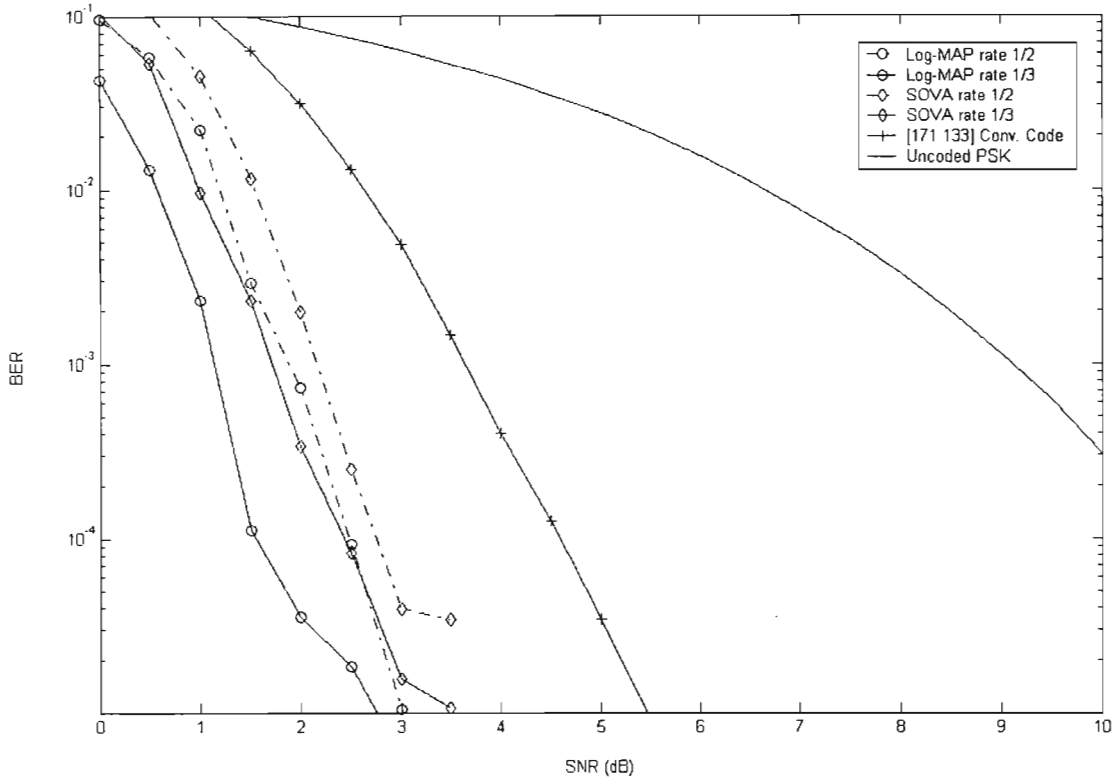


Figure 4.11: Performance comparison for error coding techniques

Figure 4.12 shows the effect of iterations on performance for a SNR = 1 dB. The same parameters as the ones in the simulations for Figure 4.11 were used. As can be seen, the performance improves for each iteration, however after nine iterations the BER does not improve significantly for all decoding algorithms. The results of Woodard and Hanzo in [24] supports this, the improvements in their results from eight to sixteen iterations is 0.1 dB. For complexity reasons it may therefore be suitable to use no more than nine iterations. As with Figure 4.11, the MAP algorithms outperform the SOVA algorithms, and the un-punctured codes outperform the punctured codes. The un-punctured codes tend to converge quicker than the punctured codes.

The length of the frames has a large effect on the performance. The original paper on turbo codes by Berrou, Glavieux, and Thitimajshima [28], show near-Shannon limit results for large frame lengths. There is a disadvantage in using such large frame lengths in that it results in large delays. However, smaller frame lengths still result in good performance [24]. Dolinar *et al.* do a more in depth investigation into the effect of the frame length on turbo code performance in [33].

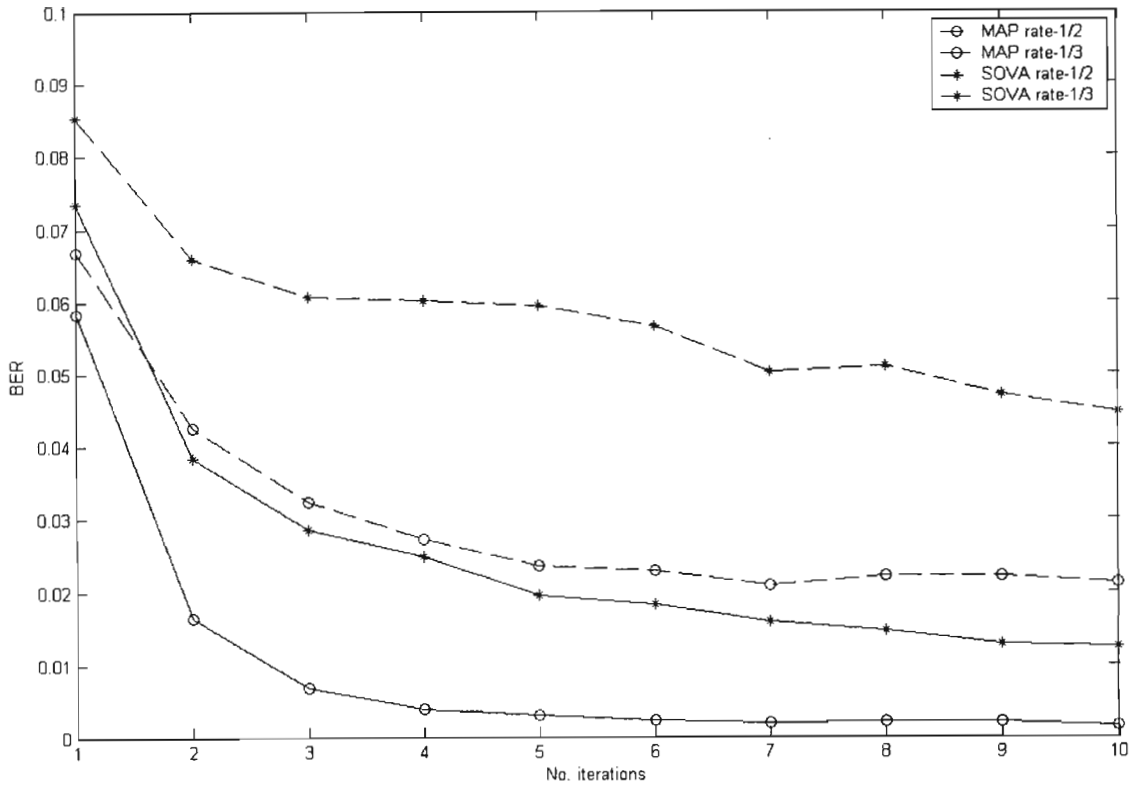


Figure 4.12: Convergence of various decoding algorithms

The interleaver can also have an effect on the performance. As mentioned above, it was suggested in [23] that the size of the interleaver is made large. [34] supports this, stating that the performance of turbo codes using short length interleavers may be worse than that of convolutional codes, and the superior performance is due to long interleavers. A method of improving the performance is to maximise the minimum free distance of the code [34], however the resulting interleavers may not necessarily be optimum [24]. Another method proposed to improve performance is odd-even separation [35]. It is shown in [24] that a 12x16 block interleaver has a lower performance than interleavers using odd numbers of rows and columns, in this instance 13x15 and 11x17 interleavers. It is also shown that the more rectangular of the two (the 11x17 interleaver) has a lower performance than the 13x15 interleaver.

This section has shown that lower-rate codes exhibit better performance, and that codes utilising MAP-algorithms show a higher performance over those using SOVA decoders. It can also be seen that the improvement in the performance after 9 iterations is low, and can be considered negligible when traded off against complexity. The effects of frame length and interleaver design were discussed briefly.

4.5 Conclusion

This chapter described a number of error coding schemes. The focus of the chapter was on convolutional codes and turbo coding schemes, as they from an integral part of iterative multiuser detection. The decoding methods for the convolutional codes and turbo codes were derived. The performances of the SOVA decoder and the MAP decoder were simulated and compared. It can be seen that the MAP algorithm shows a slightly superior performance to that of the SOVA decoder, and that rate-1/3 codes exhibit superior performance to that of rate-1/2 codes. Both iterative decoding techniques converged within 9 iterations.

CHAPTER 5 BLIND ITERATIVE MUD WITH ERROR CODING

This chapter gives an overview of iterative multiuser detection. Selected iterative detectors that have been proposed in literature are discussed, and their performance compared. Iterative adaptations of the blind detector are summarised. Previous work on blind iterative detectors is described. The proposed blind iterative MUD, meant for data communication applications in both rural and urban environments, is presented. The differences between the proposed detector and existing detectors are described, and modifications that may be made to the proposed detector are given. An analytical model of blind iterative detection, using SOVA decoders is given.

5.1 Iterative MUD

5.1.1 General

Figure 5.1 shows a simplified block diagram of an iterative CDMA decoder. The soft output CDMA decoder block is described in more detail in Section 5.1.2. Depending on the method used for the CDMA detector, feed-forward and feedback filters may need to be employed. The functioning of the interleavers and de-interleavers is described in Section 4.2. The figure only shows one interleaver, a SISO decoder and a de-interleaver. In reality, after the received signal **r** has passed through the soft output CDMA decoder, it is passed through a bank of k parallel de-interleavers, SISO decoders, interleavers and filters (not shown) where necessary, where k is the number of users. Essentially, each user's signal has its own path of de-interleavers (denoted by π^{-1}), decoders and interleavers (denoted by π) after the soft-output MUD block has processed it. This path of a single user's signal is what is represented in Figure 5.1.

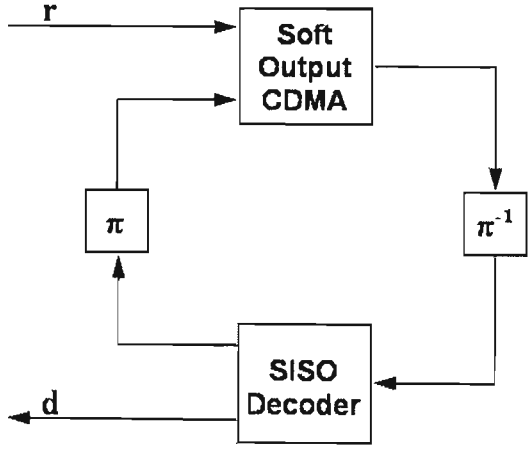


Figure 5.1: General block diagram of an iterative decoder

5.1.2 Soft-Output CDMA Decoding

Many types of SISO CDMA decoders for iterative decoding have appeared in literature, the most common are based on MMSE [36-42] or IC [41-46] detectors. Iterative detectors based on the decorrelating multiuser detector [40] and single user detectors [47] have also been considered. Due to the numerous variations in the implementation of the soft-output CDMA decoder, it is not practical to give an overview of them all. A general description of the workings of the iterative detector, as shown in Figure 5.1, will be given. A brief description of selected iterative detectors, whose methods were considered to be used as the basis of this project, will be given.

The soft output CDMA decoder accepts the received signal **r**, and generates soft outputs, which are passed through the de-interleaver to the SISO decoder, which can be either a SOVA or MAP decoder, operating as described in Section 4.4.2. It accepts soft-output decisions from the CDMA decoder and outputs its own soft-decisions, which are interleaved to be processed for the next iteration. After the required number of iterations have been completed, the hard decision **d** is made. Iterative decoders converge to their final value ranging between 3 iterations [48] and 10 iterations [41].

5.1.3 Selected Iterative Detector Structures

The receiver structure proposed in [47] is similar to the one shown in Figure 5.1, where the SISO block is a MAP decoder, and the Soft Output CDMA block is a 'CDMA MAP' decoder. The difference between the two is that a MAP decoder outputs message symbols after accepting coded symbols, whereas the CDMA MAP decoder accepts the spread symbols as an initial input – it needs to be able to take into account the CDMA channel. The soft output of the CDMA MAP decoder is

$$R_{CDMA} = \left\{ p(\underline{y} \mid d_j) : j = 1, \cdots LK \right\}, \tag{5.1}$$

where y is the received signal, d is the coded message, L is the number of symbols, K is the number of users, and

$$p(\underline{y} \mid d_j) = p(\underline{y}, d_j) / P(d_j), \tag{5.2}$$

where $P(d_j)$ is the a priori information about the symbol d_j and is set to 0.5 for the first iteration, for both $d_j = +1$ and $d_j = -1$, where

$$p(\underline{y}, d_j) = \sum_{\underline{d}_{j-K+2}^{j} \in \{-1,+1\}^{K-2}} p(\underline{y}, \underline{d}_{j-K+2}^{j}), \tag{5.3}$$

and

$$p(\underline{y}, \underline{d}_{j-K+2}^{j}) = p(\underline{y}_{1}^{j}, \underline{d}_{j-K+2}^{j}) p(\underline{y}_{j+1}^{LK} \mid d_{j-K+2}^{j}). \tag{5.4}$$

The interleavers and the MAP decoder function as described in Sections 4.2 and 4.4.2 respectively.

Figure 5.2 shows the block diagram of the proposed receiver in [45]. Here the 'APP' blocks represent MAP decoders, and they, along with the interleavers and de-interleavers, function as described above. The 'tanh' functions generate soft-decisions with which the cancelled received signal can be calculated. Each individual decoder processes the received signal and generates soft-decision values of the transmitted coded symbols $\mathbf{d}_k[j]$. Once the soft estimates $\hat{\mathbf{d}}_k[j]$ of the symbols have been generated by the tanh functions, a cancelled received signal for user k at time j can be calculated [45]:

$$\mathbf{r}_{k,j} = \mathbf{S}\mathbf{d} - \mathbf{S}\hat{\mathbf{d}}_{k,j} + \mathbf{z}, \tag{5.5}$$

where $\mathbf{S}$ is a matrix of the spreading sequences, $\mathbf{z}$ is the noise term, and $\hat{\mathbf{d}}_{k,j} = [\hat{\mathbf{d}}_1[1], \cdots, \hat{\mathbf{d}}_{k-1}[j], 0, \hat{\mathbf{d}}_{k+1}[j], \cdots, \hat{\mathbf{d}}_k[K]]$, although typically only estimates around the symbol of interest are required.

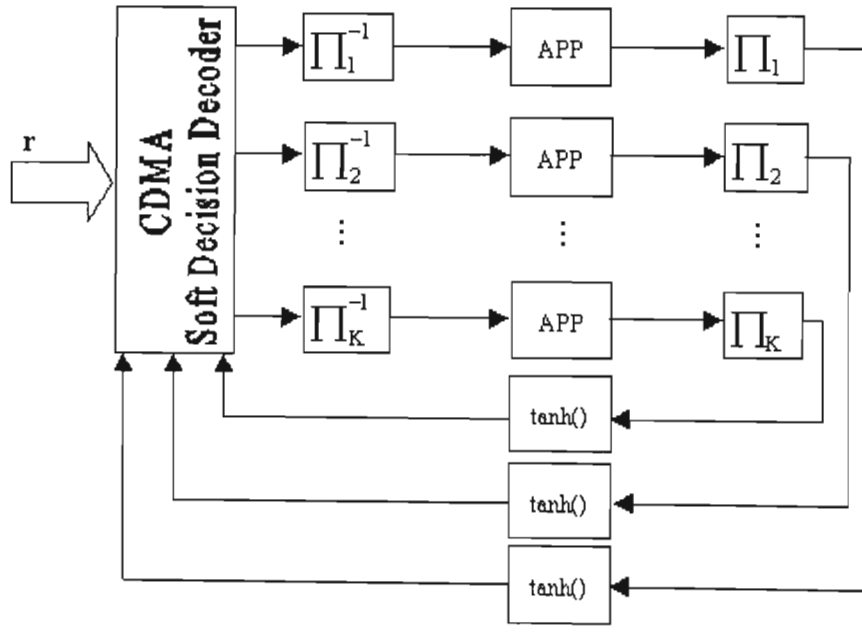


Figure 5.2: Block diagram of joint iterative decoder [45]

Residual interference is suppressed through the use of linear filters (part of the CDMA soft decision decoder), in the general form of

$$\mathbf{y}_k[j] = \mathbf{w}'_{k,j}\mathbf{r}_{k,j}, \tag{5.6}$$

where $\mathbf{w}_{k,j}$ is the filter for the kth user at time j. The interference canceller ignores the structure of mutual interference, and uses the spreading sequence so $\mathbf{w}_{k,j} = \mathbf{s}_{k,j}$. The MMSE approach uses a more sophisticated filter

$$\mathbf{w}_{k,j} = \frac{P_k}{1 + P_k \mathbf{s}'_{k,j}\mathbf{K}^{-1}_{k,j}\mathbf{s}_{k,j}}\mathbf{K}^{-1}_{k,j}\mathbf{s}_{k,j}, \tag{5.7}$$

where $\mathbf{K}_{k,j} = \mathbf{S}'_{k,j}\mathbf{D}_k\mathbf{S}_{k,j} + \sigma^2\mathbf{I}$, $\mathbf{D}_k$ is a diagonal matrix of the residual power of interfering users, and $\mathbf{S}_{k,j} = [\mathbf{s}_{1,1}, \cdots, \mathbf{s}_{k-1,j}, \mathbf{s}_{k+1,j}, \cdots, \mathbf{s}_{K,L}]$. As the MMSE filter is heavily dependant on the signal-to-noise ratio, the normalisation factor, $P_k / \left(1 + P_k \mathbf{s}'_{k,j}\mathbf{K}^{-1}_{k,j}\mathbf{s}_{k,j}\right)$, can be dropped, reducing the filter to

$$\mathbf{w}_{k,j} = \mathbf{K}^{-1}_{k,j}\mathbf{s}_{k,j}. \tag{5.8}$$

By applying the stationary inversion, $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \left(\mathbf{M}\mathbf{x}^{(n)} - \mathbf{b}\right)$, to $\mathbf{K}^{-1}_{k,j}$, a multistage filter implementation is arrived at

$$\mathbf{w}^{(n+1)}_{k,j} = \mathbf{\Phi}_n(\mathbf{K})\mathbf{s}_{k,j}, \tag{5.9}$$

where each stage in the filter consists of a simple matrix multiplication $\mathbf{K}_{k,j}\mathbf{w}^{(n)}_{k,j}$ [45].

Figure 5.3 shows a block diagram of the receiver structure proposed in [48]. The receiver takes the matched filter output, $\underline{y}_t$, and generates conditional channel probabilities $p(\underline{y}_t \mid \underline{d}_t)$, which are multivariate Gaussian conditional probabilities [20]. The marginal probabilities for the kth decoder, $p\left(\underline{y}_t \mid \underline{d}^{(k)}_t\right)$, are calculated by the metric generator. The single-user SISO MAP decoders accept the output from the metric generator, and output *a posteriori* information back to the metric generator as *a priori* information for the next iteration, as described above. The decision rule for the metric generator is [48]:

$$\begin{aligned} \underline{d}_t &= \arg\max_{\underline{d}_t} \frac{p(\underline{d}_t, \underline{y}_t)}{p(\underline{y}_t)} \\ &= \arg\max_{\underline{d}_t} p(\underline{y}_t \mid \underline{d}_t) \end{aligned} \tag{5.10}$$

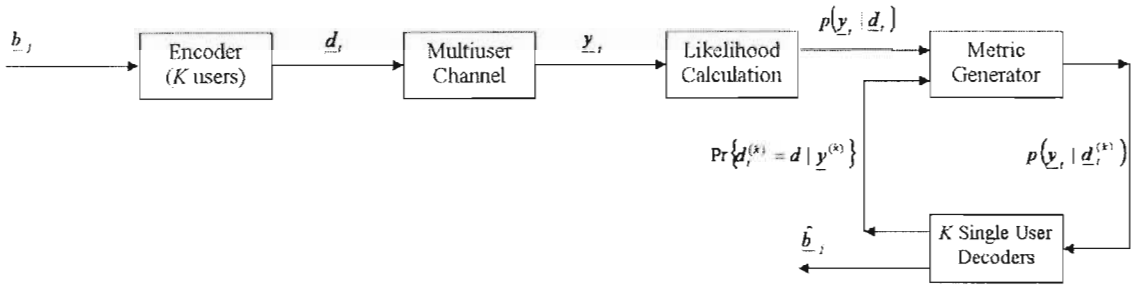
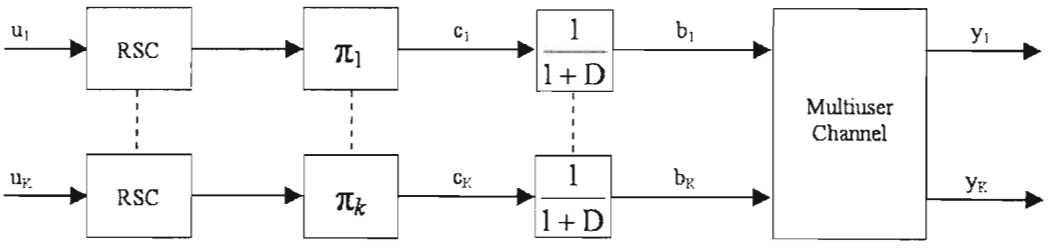
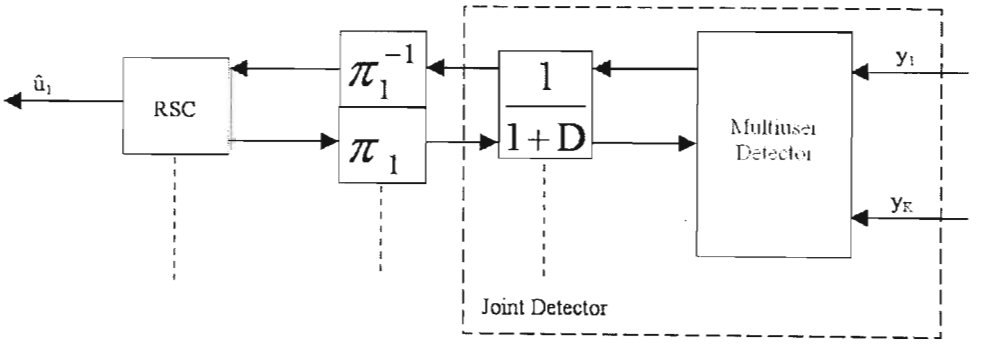


Figure 5.3: Block diagram proposed by Reed *et al* [48]

The block diagram in Figure 5.4 shows the system proposed in [49]. This method employs a serially concatenated convolutional code, which consists of a RSC encoder, an interleaver, and a differential encoder, for each user. The receiver utilises a bank of K matched filters to decode the CDMA signal. The received signal is iterated between the RSC decoder, the differential decoder and the multiuser detector [49].



(a) Transmitter



(b) Receiver

Figure 5.4: Joint detector proposed by Xia and Ryan [49]

The table below gives a summary of the performance of selected iterative decoders, giving the SNR at which a P_e of 10^{-4} was achieved and the number of iterations for the detector to converge (where available):

Table 5.1: Iterative detector performances

	SNR where a $P_e = 10^{-4}$ achieved	Convergence
Darwood *et al* [46]	11 dB	7
Alexander *et al* [47]	5 dB	
Reed *at al* [48]	4 dB	3
Xia and Ryan [49]	≈2.3 dB	
Grant and Alexander [57]	≈5.6 dB	5

5.2 Iterative Blind Detectors

Derryberry *et al* proposed a method whereby the blind adaptive algorithm itself is modified to be iterative [50]. The blind adaptive algorithm as described in Section 3.7.2 updates the weight vector once per symbol interval, which could lead to many symbols being processed while the weight vector is slowly converging to the solution [50]. The iterative blind adaptive algorithm reduces the time taken to converge via multiple updates of the weight vector per symbol. This algorithm is iterative in that each update of the weight vector corresponds to processing the same observed signal sample vector with different observed interference vectors [50]. There is also mention of the fact that the data symbols can be broken into smaller 'sub-symbols', however this also reduces the processing gain, which is not desirable [50].

Whilst this method, when used with an error coding scheme, satisfies the purpose of this thesis, it does not meet the objectives of having an integrated CDMA and FEC coding architecture. The iterative blind adaptive algorithm combined with an iterative error-coding process proves computationally complex.

A Gaussian Mixture Model based on Expectation Maximization (EM) channel estimation is proposed in [51] and [52], and is also described in [53] for space-time coding systems. Again, this method is not suitable due to the fact that the spreading code did not need to be known for detection, yet the channel needed to be accurately estimated. A system using the blind adaptive algorithm with an integrated CDMA and FEC coding architecture would reduce the computational complexity of the CDMA receiver significantly.

5.3 Blind Iterative MUD with Error Coding

5.3.1 Previous Work

The detector shown if Figure 5.5 is a blind iterative decision-feedback, using the MOE criterion, proposed in [54]. From the figure $\mathbf{F}^m$ and $\mathbf{B}^m$ denote the feedforward and feedback filters, in the mth iteration, respectively, and ξ_k^m denotes the interference that is assumed to be Gaussian, and A_k^m denotes the amplitude. The priors are calculated from the estimates of A_k^m and $\text{var}\left(\xi_k^m\right)$, which are gained from time averages of $d_k^*(i)y_k^m(i)$ and $\left[y_k^m(i) - A_k^m d_k(i)\right]^2$ [54]. Training sequences would be required at the receiver to obtain the time averages, and the blind adaptive MOE method, described in Section 3.7 and [1,17-19], is used to generate symbol decisions that can replace the training sequences [54].

The received signal is given by [54]:

$$\mathbf{r}(i) = \mathbf{A}\mathbf{d}(i) + \mathbf{n}(i), \tag{5.11}$$

where $\mathbf{A}$ is the channel matrix, $\mathbf{d}(i)$ is the vector symbols contributing to $\mathbf{r}(i)$ and $\mathbf{n}(i)$ is the additive white Gaussian noise vector. The blind adaptive MOE algorithm produces the decision statistics [54]

$$\mathbf{d}^1(i) = [\mathbf{A} + \mathbf{X}(i)]^H\mathbf{r}(i), \tag{5.12}$$

where $\mathbf{X}(i)$ satisfies the constraint $\mathbf{A}^H\mathbf{X}(i)$, where $\mathbf{X}(i)$ is obtained as explained in Section 3.7. The hard decisions of the estimate $\mathbf{d}^1(i)$, $\mathbf{d}'(i) = \text{dec}[\mathbf{d}(i)]$, are then used in estimating A_k^1 and $\text{var}\left(\xi_k^1\right)$. For the first iteration the feedforward and feedback filters are given by $\mathbf{F}^1 = [\mathbf{A} + \mathbf{X}]^H$ and $\mathbf{B}^1 = \mathbf{0}$, respectively [54].

The iterative algorithm is implemented as follows:

- For the first iteration, the MOE algorithm is used to compute $\mathbf{X}$ and form the decision statistics $\mathbf{d}^1(i) = [\mathbf{A} + \mathbf{X}(i)]^H\mathbf{r}(i)$, from which the hard decisions are taken and used to estimate A_k^1 and $\text{var}\left(\xi_k^1\right)$. The priors are then computed using Gaussian assumption on ξ_k^1, de-interleaved and input into the MAP decoder.

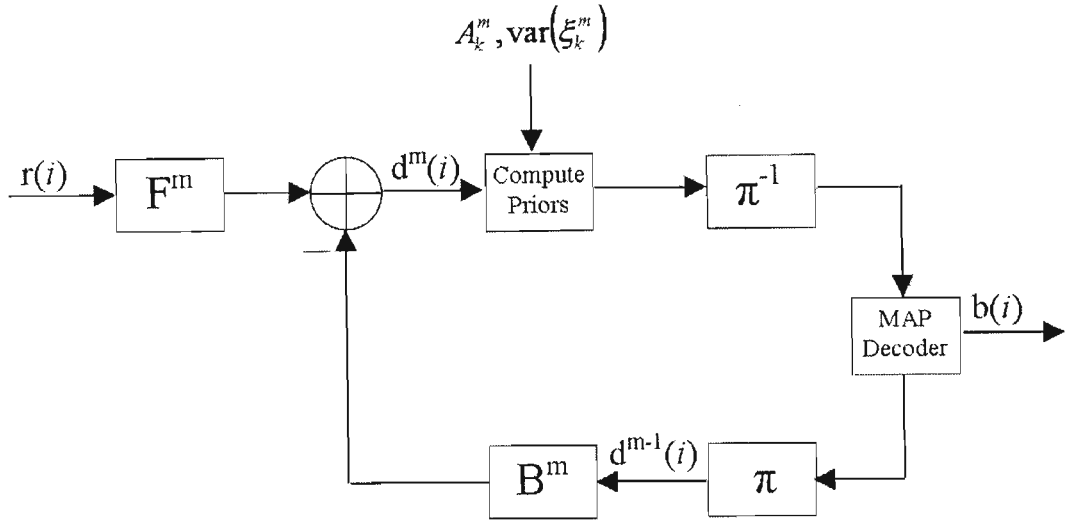


Figure 5.5: Blind iterative receiver proposed by Lim *et al* [54]

- For subsequent iterations, the *a posteriori* probabilities of the coded bits are interleaved and soft estimates are formed from the APP's to compute the necessary time-averaged correlation matrices [55], which in turn are used to compute $\mathbf{F}^m$ and $\mathbf{B}^m$. Soft estimates of the coded symbols $\mathbf{d}^m(i) = \mathbf{F}^m\mathbf{r}(i) - \mathbf{B}\mathbf{d}^{m-1}(i)$ and hard decisions $\mathbf{d}'(i) = \mathrm{dec}[\mathbf{d}(i)]$ are formed. A_k^m and $\mathrm{var}(\xi_k^m)$ are obtained from $\mathbf{d}'(i)$, priors are computed, de-interleaved and inputted into the MAP decoder.

5.3.2 New Work

The proposed receiver is a combination of the receivers described in [54] and [48], which are described in Sections 5.3.1 and 5.1.3 respectively. The proposed detector is shown in Figure 5.6. The message is FEC coded with a convolutional code, then passed through an interleaver. The output of the interleaver is then spread and modulated before being transmitted across the multiuser channel. A SOVA algorithm replaces the MAP algorithm used in [48] and [54] due to the fact that although its performance is slightly worse, the reduction in complexity makes this negligible.

The MOE criterion for blind detection was decided upon, as it is the more common of the blind techniques. The MOE detector makes the initial estimates as before. The interleaver is used to 'distribute' any further burst errors that may occur. Once the priors have been computed, either by using the hard decisions of the MOE detector to estimate A_k^1 and $\mathrm{var}(\xi_k^1)$, or they may initially set to P(1) = P(-1) = 0.5, which is computationally less complex. The SOVA decoder may then decode the message. The feedback filter, **B**, operates as above in that it has an initial value of zero that may be altered as required, and may simply be used as a gain to strengthen the signal if it is not required. The primary function of the feedback filter is to ensure that the

algorithm remains stable. The integration of the MUD and FEC decoders comes in that the MOE detector provides information about the channel to compute the priors used by the SOVA decoder, and to determine the filter **B** as required.

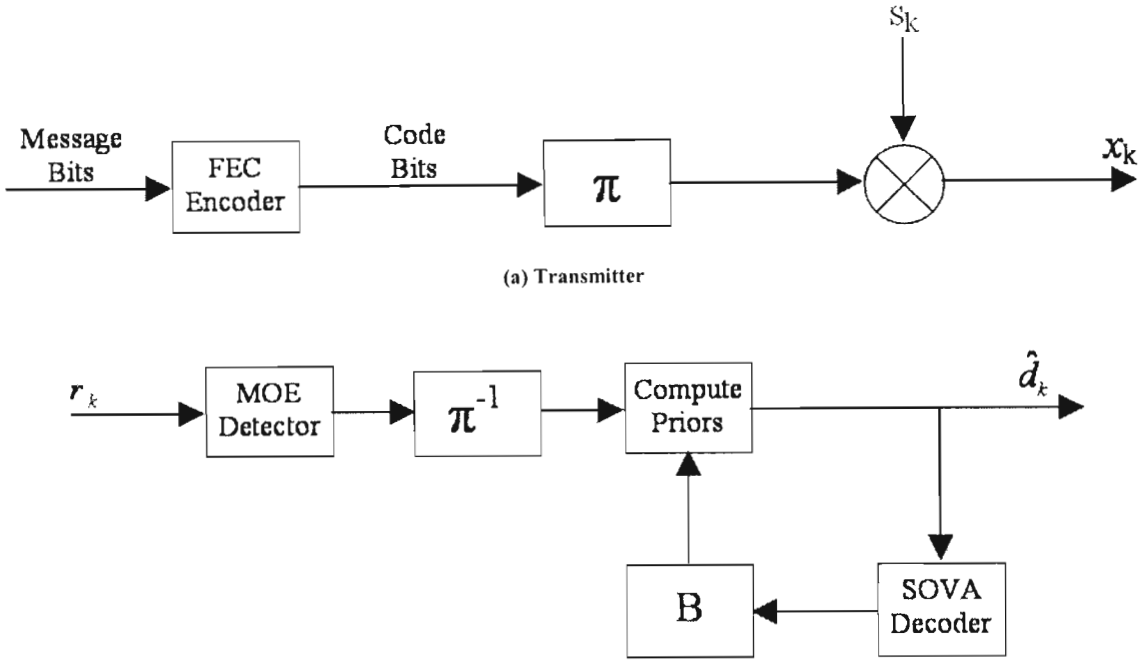


Figure 5.6: Proposed blind iterative MUD with FEC

5.3.3 Analysis

The analysis of the proposed detector will be described using the equations for the blind detector using the MOE criterion. It may be noted, that as the blind detector approximates the MMSE detector in terms of performance, the equations to model an MMSE detector may be used.

The output SIR of the blind detector is given by Equations 3.15 and 3.16. The BER at the output of the blind detector is given by Equation 3.17.

The multi-user variance can be derived as in [45]. For the case of equal transmitted power the residual interference variance $\sigma_k^2 \equiv \sigma^2$ and the symbol variance $\sigma_{d,k}^2 = \sigma_d^2$ are both independent of the channel population [45]. The variance transfers are given by [45]:

$$\sigma^2 = f_{\text{mud}}(\sigma^2, P, N_0) \tag{5.13}$$

$$\sigma_d^2 = f_{\text{dec}}(\sigma^2), \tag{5.14}$$

where f_{mud} and f_{dec} are the transfer functions for the MUD and error decoder respectively. The variance transfer analysis is hence reduced to a one-dimensional dynamical analysis, which is derived in [56] for iterative detectors using interference cancellation. This method still applies for performance analysis if the users have unequal received powers [45].

For the case where the power is unequal the variance transfer can be analysed in a similar manner to the equal power case by constructing effective variance transfer functions for the decoders. The effective variance transfer function has the general form [45]:

$$\sigma_{eff}^2 = \frac{N_0}{2P} + \frac{K_{eff}}{N}\sigma_{d,eff}^2, \tag{5.15}$$

where σ_{eff}^2 is the interference variance normalised with respect to the power level P, N is the spreading gain, and $K_{eff} = \frac{1}{P}\sum_{l=1}^{J} K_l P_l$ is the effective number of users for unequal power, where J denotes time. The effective symbol variance is denoted by $\sigma_{d,eff}^2$, and can take on values between 0 and 1 [45]. It is obtained by getting the variance transfer function for the FEC, f_{dec}^*, via numerical experimentation, from which the symbol variance can be calculated as:

$$\sigma_{d,j}^2 = f_{dec}^*\left(\frac{P}{P_j}\sigma_{eff}^2\right), \tag{5.16}$$

where $j \in \{1, \cdots, J\}$. The effective symbol variance can then be calculated as [45]:

$$\sigma_{d,eff}^2 = \sum_{j=1}^{J} \alpha_j \left(\frac{P\sigma^2\sigma_{d,j}^2}{P\sigma^2 + P_j\sigma_{d,j}^2}\right), \tag{5.17}$$

where α_j is the weighting factor, and $\sum_{j=1}^{J} \alpha_j = 1$.

The variance transformation over each iteration for BPSK modulation is also derived in [46, 56] as follows; the noise variance at the output of a Viterbi decoder with a hard output assuming an input variance σ_{in}^2 is given by [46, 57]:

$$\sigma_{FEC}^2 \approx 4d_{free}Q\left(\sqrt{\frac{d_{free}}{\sigma_{in}^2}}\right), \tag{5.18}$$

At the next iteration the total input noise to the joint detector becomes [46, 56]:

$$\sigma_{JD}^2 = \beta\sigma_{in}^2 + N_0, \tag{5.19}$$

where $\beta = K/N$ is the normalised load. Combining Equations (5.18) and (5.19), and taking into account the BPSK modulation, the variance transfer can be given by the following relationship [46,56]:

$$\sigma_{m+1}^2 = 4d_{\text{free}}Q\left(\sqrt{\frac{2d_{\text{free}}}{\beta\sigma_m^2 + N_0}}\right), \tag{5.20}$$

where m denotes the iteration. Given that the Q-function can be defined as in Equation 3.24, the variance transformation for BPSK modulation becomes:

$$\sigma_{m+1}^2 = 2d_{\text{free}}Q\left(\sqrt{\frac{d_{\text{free}}}{\beta\sigma_m^2 + N_0}}\right). \tag{5.21}$$

The BER at the output of the receiver can be calculated from the instantaneous SIR by [58]:

$$P_e = Q\left(\sqrt{SIR}\right). \tag{5.22}$$

Applying Equation 3.24 we then get:

$$P_e = \tfrac{1}{2}erfc\left(\sqrt{SIR/2}\right). \tag{5.23}$$

The stability is analysed in [57] as follows; the derivative below is considered:

$$F'(x) = \frac{2\beta}{\sqrt{\pi}}\left(\frac{d_{free}}{\beta x + \sigma^2}\right)^{3/2}\exp\left(\frac{-d_{free}}{\beta x + \sigma^2}\right). \tag{5.24}$$

Noting that $\beta(\beta x + \sigma^2)^{-3/2} < 1$, the following condition for stability is obtained [57]:

$$0 < x < \frac{1}{\beta}\left(\frac{2d_{free}}{3\ln d_{free} - \ln \pi/4} - \sigma^2\right) \Rightarrow F'(x) < 1. \tag{5.25}$$

5.4 Conclusion

An overview of iterative multiuser detection has been given in this chapter. Selected iterative detectors were described and compared. A summary of iterative blind algorithms was given. Previous blind iterative detectors were described and the proposed detector was presented. An analytical model for the proposed detector was given.

CHAPTER 6 PERFORMANCE OF THE PROPOSED BLIND ITERATIVE DETECTOR

This chapter presents the simulated and analytical results for the detector proposed in Section 5.3.2. The simulated performance of the proposed detector is compared to the performance of the blind MOE detector, intended for data communication in both rural and urban environments; however the simulation will focus more on the rural environment. The effects of various channel parameters on the performance of the proposed detector are simulated and discussed.

Simulations were performed with a frame length of 1500 bits and a spreading code of length 31, with no multipaths and perfect power control, unless stated. The simulated channel is AWGN. The plots are the averaged results from 150 independent simulations, unless otherwise stated. The simulated interleaver is a simple block interleaver, and a (2, 1, 1) convolutional code with generator matrices $(g_1, g_2) = (1\ 1\ 1,\ 1\ 0\ 1)$ was used, unless otherwise stated. Whilst the simulations employ an AWGN channel, the system should also be able to operate under conditions with burst errors, hence the interleaver is included. The initial *a priori* information was taken as $P(1) = P(-1) = 0.5$.

The simulated channel is a K x L_p+1 matrix, where K is the number of users, and L_p is the number of multipaths. The first column contains unity values, and the remainder of the values are less than unity and randomly generated (the first column corresponds to the main transmission path, and the other columns correspond to the multipaths). The near-far value is taken with respect to the desired user's strength. The desired user's message is passed through the channel, using Matlab's 'filter' function. The cross-correlation of the interfering users is also passed through the filter, and then added to the desired user's signal along with AWGN, to take into account MAI and channel noise respectively. One parameter i.e. SNR, users, multipaths, or the power of the interfering users in relative to the desired user (the near-far effect), can be varied at a time.

The analytical results were obtained by using the equations in Section 5.3.3. The BER was calculated from the instantaneous SIR as in Equation 5.8. The variance transfer function for the SOVA decoder was assumed to be equivalent to that of the Viterbi decoder, given by Equation 5.6. The SIR at the output of the MOE detector can be calculated as in Equations 3.15 and 3.16. This will then be the input SIR to the "prior calculation" block.

The legends of the graphs are as follows; "Iterative" denotes simulations of the proposed detector, "Analytical" denotes the analysis of the proposed detector, "Blind MOE" denotes the

performance of a blind MUD using the MOE criterion, without any error coding, as simulated in Section 3.8.2. The performances of the detectors simulated in 3.8.2 were corroborated with the performances appearing in literature. "SU Bound" denotes the single-user bound. In other cases, assume the simulation refers to the proposed blind iterative MUD if it is not explicitly stated.

6.1 General Performance and Complexity

As can be seen from Figure 6.1 the iterative detector exhibits a significantly improved BER performance over the blind MOE detector. Only the performance after 3 iterations is shown to avoid overcrowding the graph with plots, making it unreadable. The detector converges after 3 iterations, and will be discussed again later. The plots only show the SNR's up to 8 dB as the simulation plots becomes inaccurate at higher SNR's due to the low BER. These simulations were averaged over 250 independent simulations with a frame length of 4000 bits.

As can be seen, the fewer the number of users operating on the channel, the better the performance. This is to be expected due to the fact that there will be less multiple access interference to generate errors. The effect of the number of users on performance will be discussed in more detail in Section 6.2.1. The simulated plot closely follows the analytical plot for each channel population.

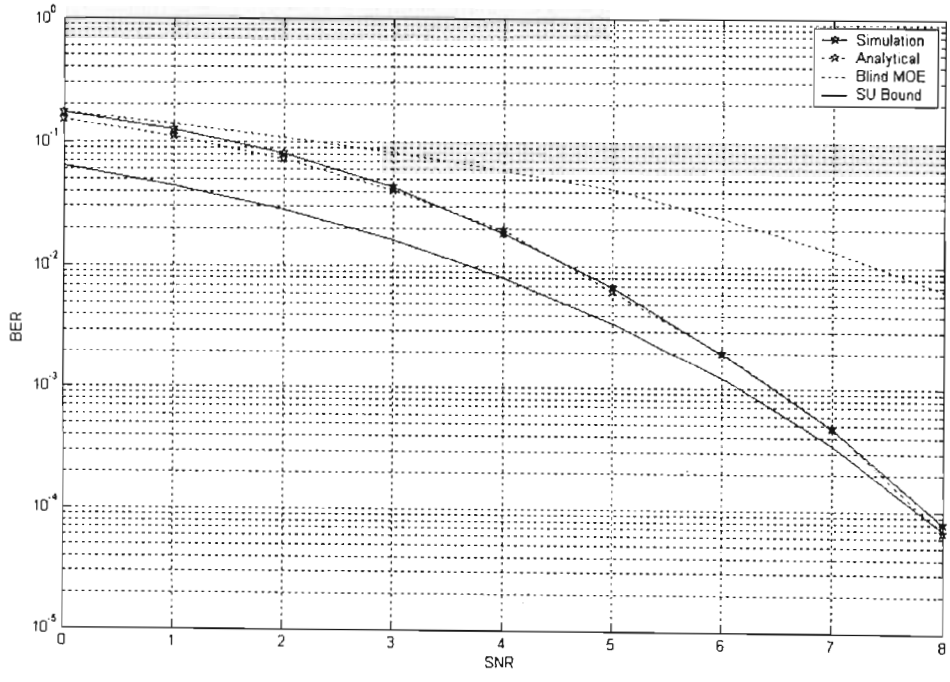
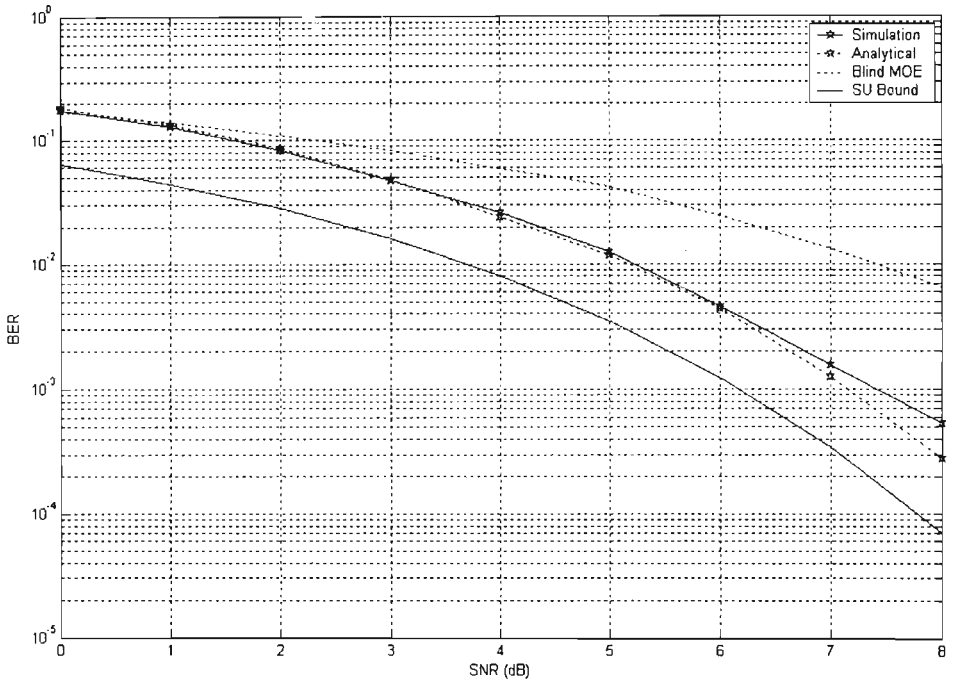
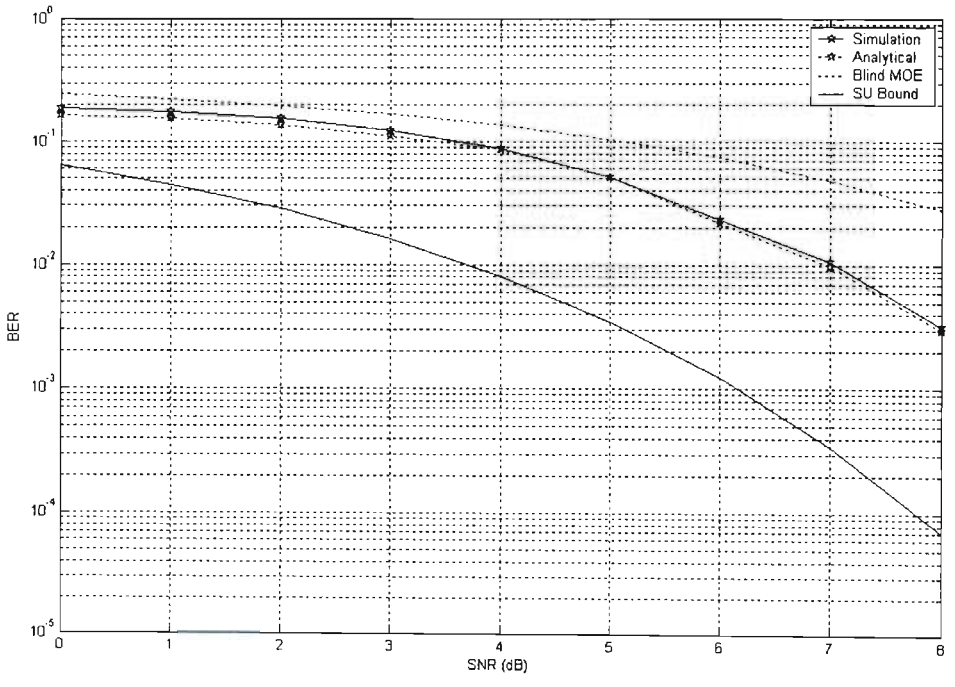


Figure 6.1 (a)



(b)



(c)

Figure 6.1: Performance after 3 iterations for (a) 2 users, (b) 5 users, and (c) 10 users

Figure 6.2 shows that the system converges to its final value, with the improvement after three iterations being negligible. Figure 6.3 shows the first three iterations, with the corresponding analytical plots. Again it can be seen that the analytical and simulated plots are very close together in value.

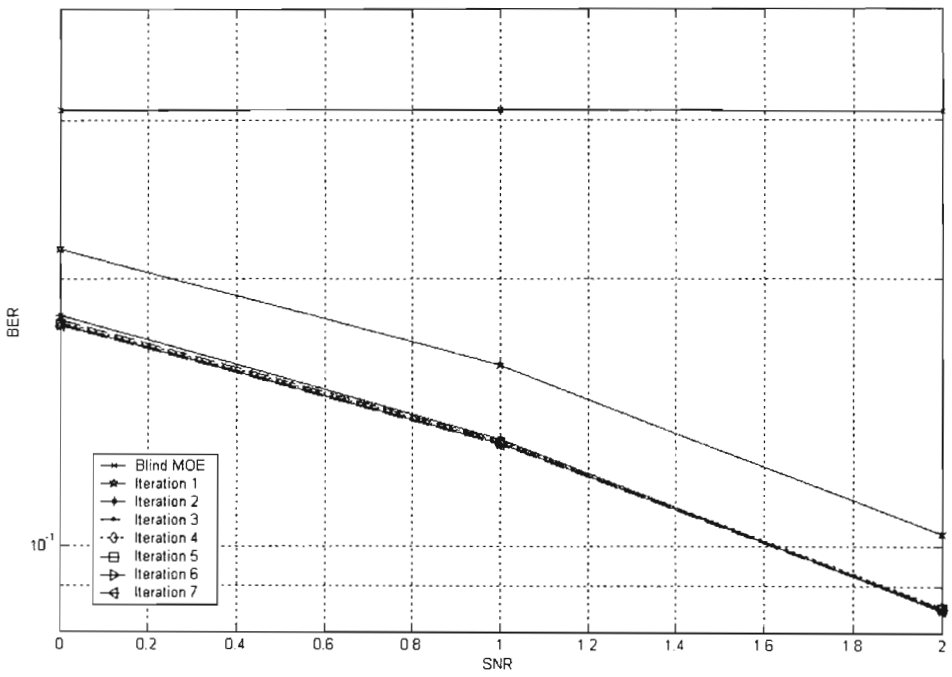


Figure 6.2: Convergence, 5 users

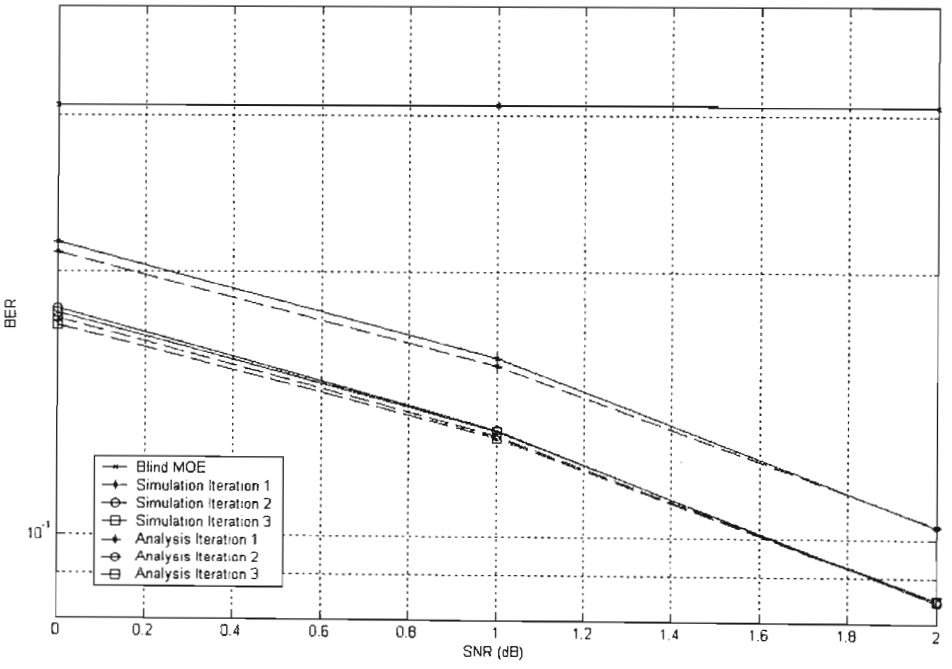


Figure 6.3: Convergence with analytical plots, 5 users

Figure 6.4 shows the effect the spreading code length has on performance. It can be seen that the performance of the proposed detector increases with an increase in code length. This corresponds to the characteristic shown in Figure 3.3 in Section 3.1.

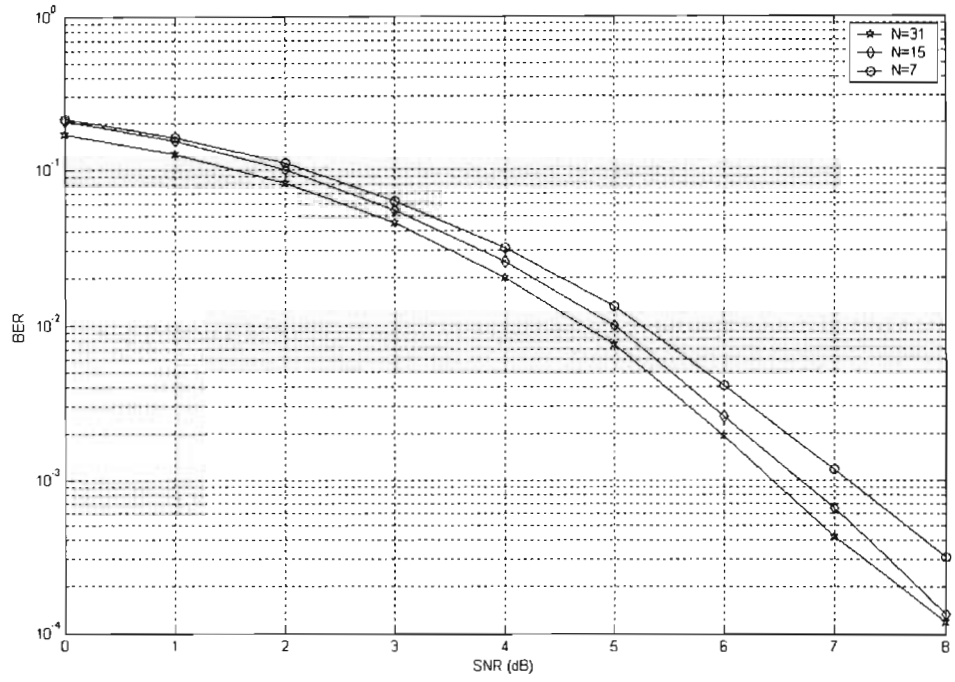


Figure 6.4: The effect of processing gain on performance, 5 users

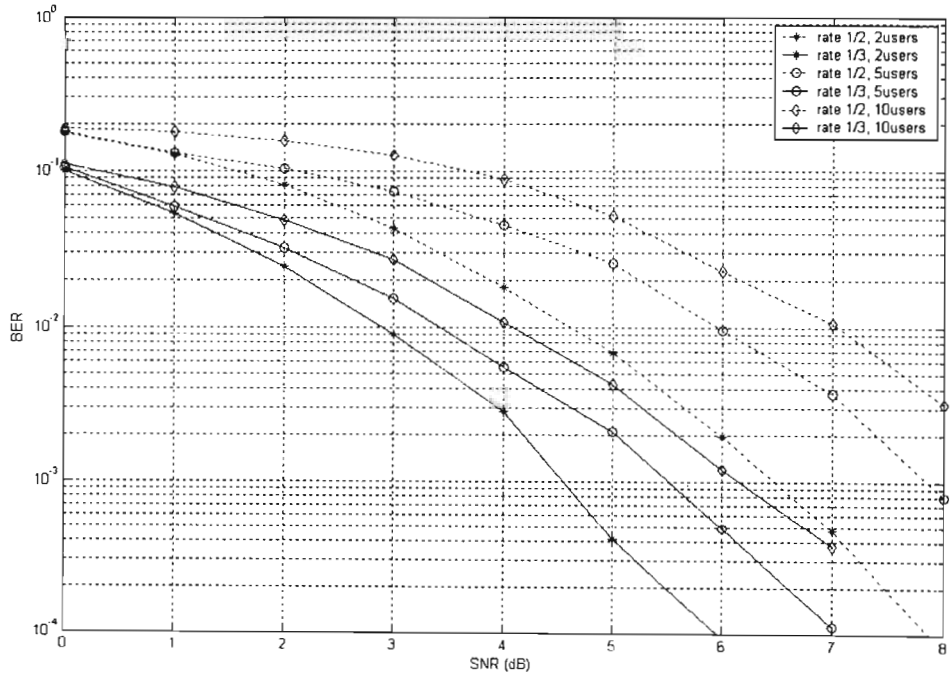


Figure 6.5: The effect of convolutional code rate on performance

Figure 6.5 shows the performance comparison of the proposed iterative MUD using a rate-1/2, as simulated for Figure 6.1, and the proposed detector using a rate-1/3 convolutional code with generator matrices (g_1, g_2, g_3) = (1 1 1, 1 0 1, 1 1 0). This is to illustrate how the system performance can be altered by changing the parameters of FEC component. The superior performance of the system employing the rate-1/3 code is clear, however the computational

complexity is increased, resulting in a longer processing time, thus a rate-1/3 code would only be used in areas with high interference or when processing time is not an issue. The results of the rate-1/3 code system is very close to those given by [54].

6.2 The Effect of Channel Parameters

6.2.1 Channel Population

Figures 6.6 and 6.7 show the effect of channel population on the iterative decoder compared to that of the blind MOE and analytical results respectively. It is apparent that with an increase in the number of users in the channel, there is a decrease in performance. It follows that the performance is decreasing due to the rising amounts of MAI due to the extra users. Again, the superior performance of the iterative decoder compared to the blind MOE detector can be seen in Figure 6.6. The simulated plots approximate the analytical plots, as is apparent in Figure 6.7. The simulated plot for the SNR=10 dB is not accurate due to the frame length being too low to get an accurate BER's. Figure 6.8 shows the convergence of the iterative detector, and the characteristic of the improvement in performance being negligible after 3 iterations is again exhibited. It should be noted that the iterative detectors performance will converge to that of other detectors as the number of users increases, as eventually the channel will become so overloaded that any signal will be drowned out.

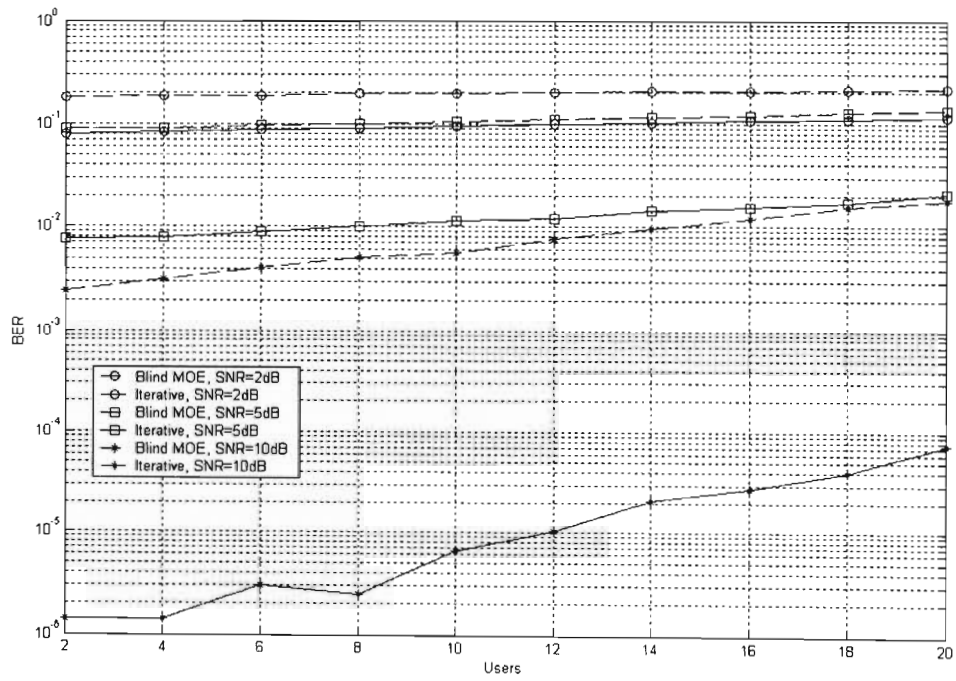


Figure 6.6: The effect of channel population on blind MOE and iterative blind detectors

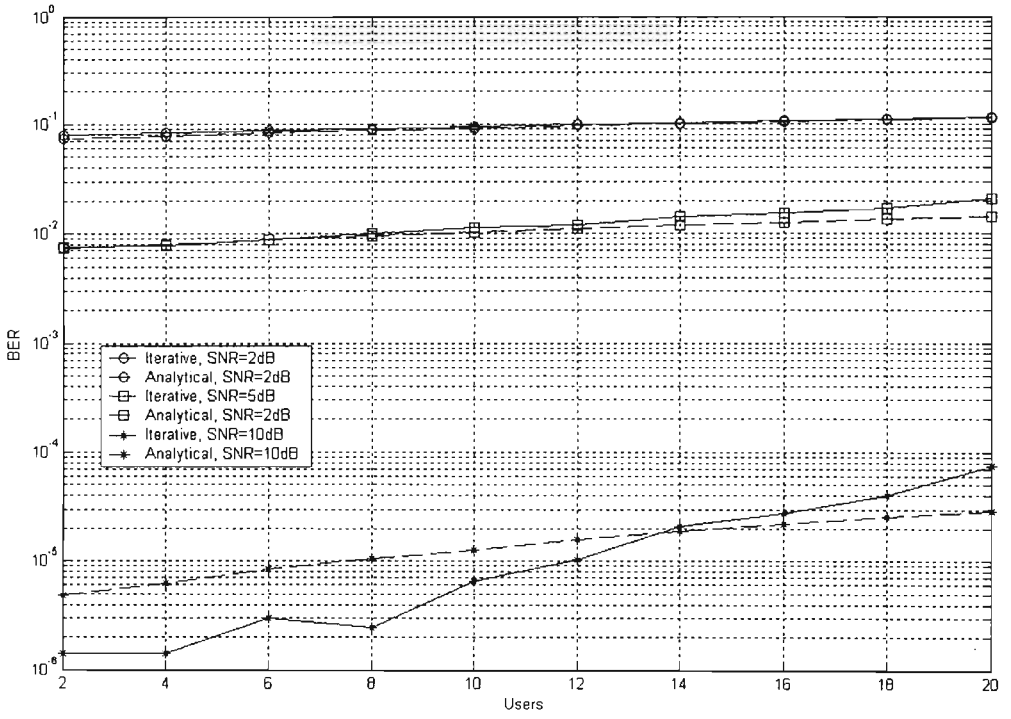


Figure 6.7: Analytical and simulated plots for the effect of channel population

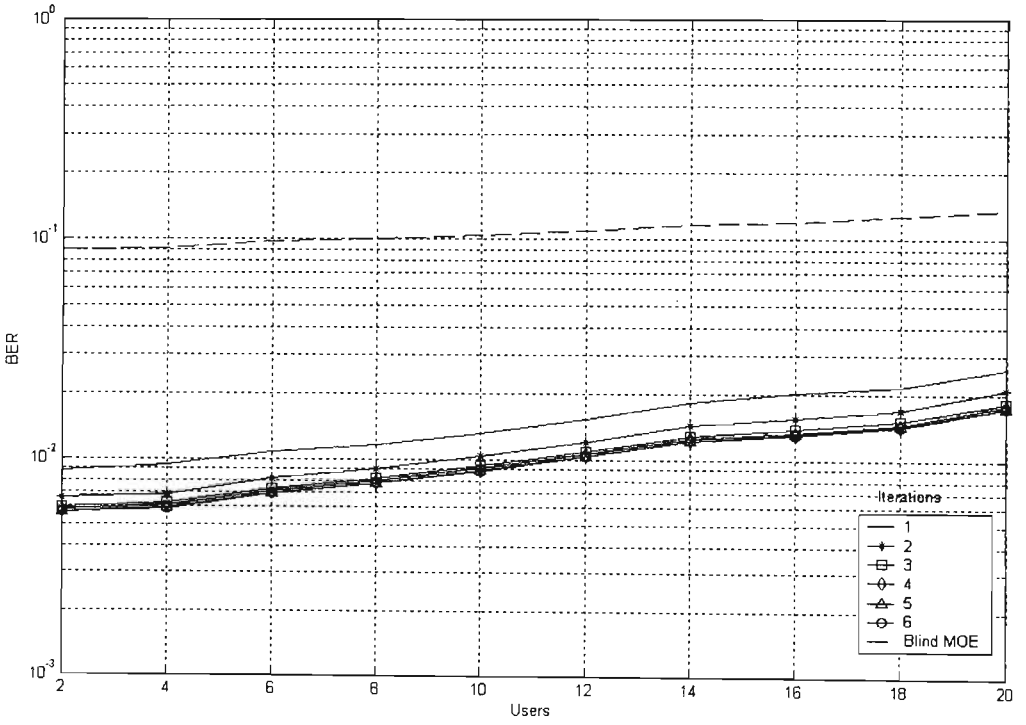
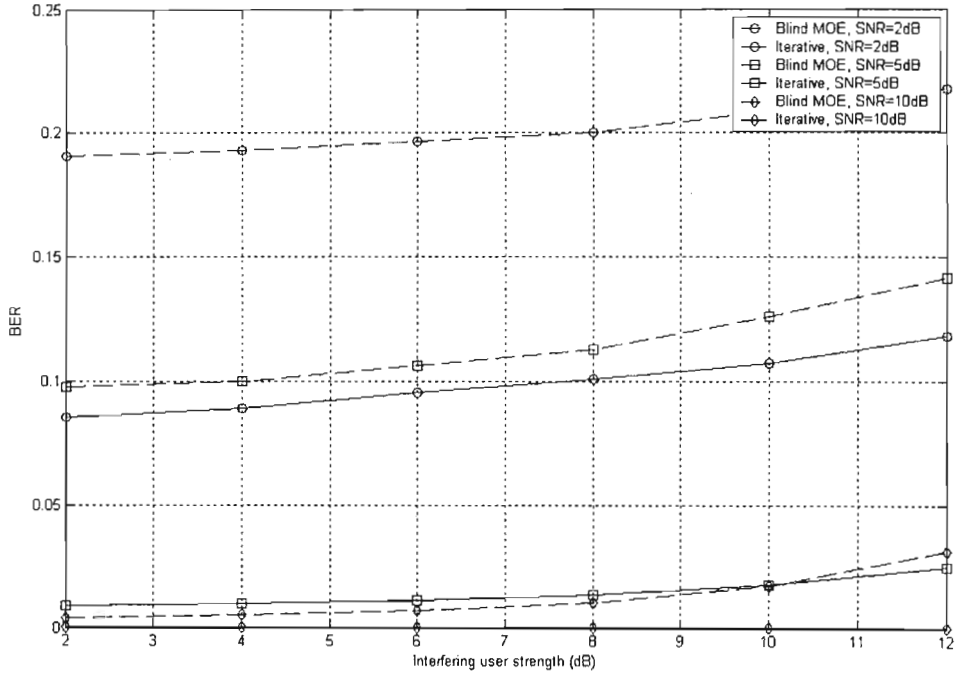


Figure 6.8: Convergence, SNR=5dB, 5 users

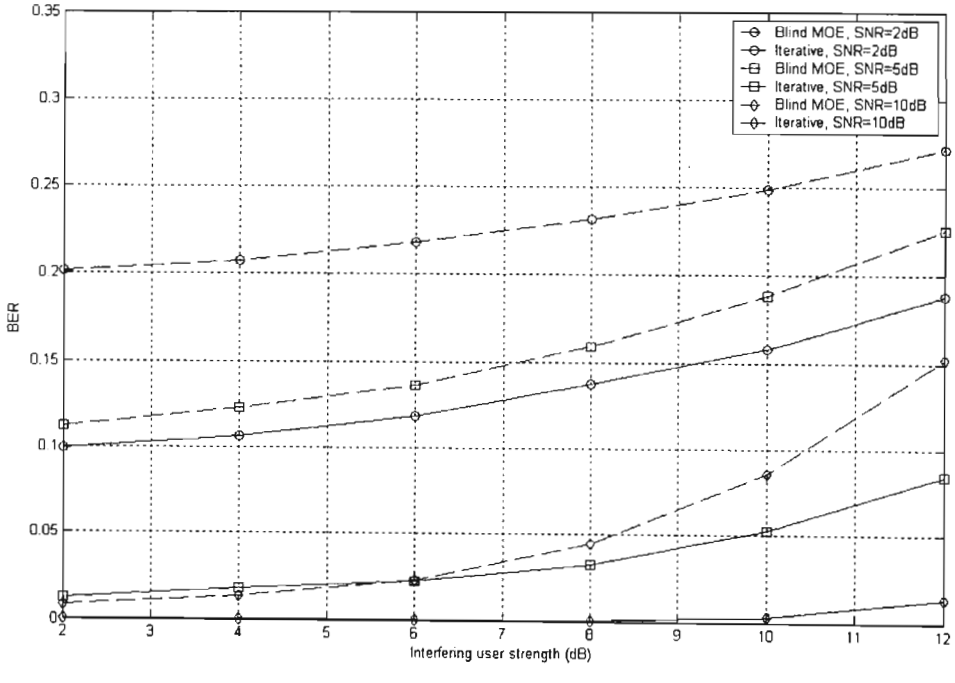
6.2.2 Near-Far Effect

Figure 6.9 shows the effect of the signal power of the interfering users relative to the desired user. As can be seen, the stronger the interfering users' signals are, the worse the performance of the desired user is. In addition, the effect is more noticeable for 10 users than for 5. Both of

these characteristics may be explained by MAI, in that an increase in an interfering user's strength, an increase in the number of users, or both, will increase the amount of MAI the desired user is subjected to. Again, the iterative decoder exhibits superior performance to the blind detector, but will converge for the same reasons as given above.



(a)



(b)

Figure 6.9: The near-far effect for (a) 5 users, and (b) 10 users

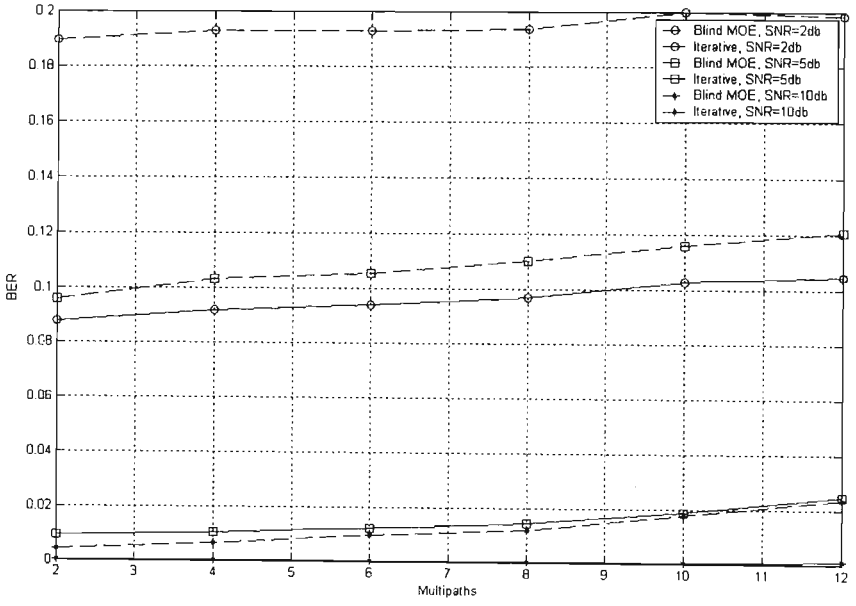
6.2.3 Multipaths

Figure 6.10 shows the effects of the number of multipaths on performance. The multipaths were simulated as in Section 3.2.1. Again it can be seen that the iterative detector exhibits an improved performance over the blind detector. The degradation in performance as the number of multipaths increases is as a result of an increased amount of interference from the extra, asynchronous signals on the channel. Again, the more users on the channel, the greater the effect the multipaths have (assuming each user has an equal number of multipaths, so for each additional user operating on the channel, there are L_p new multipaths), for the same reasons as in Section 6.2.2 above. The multipath component of the received signal can be modelled as:

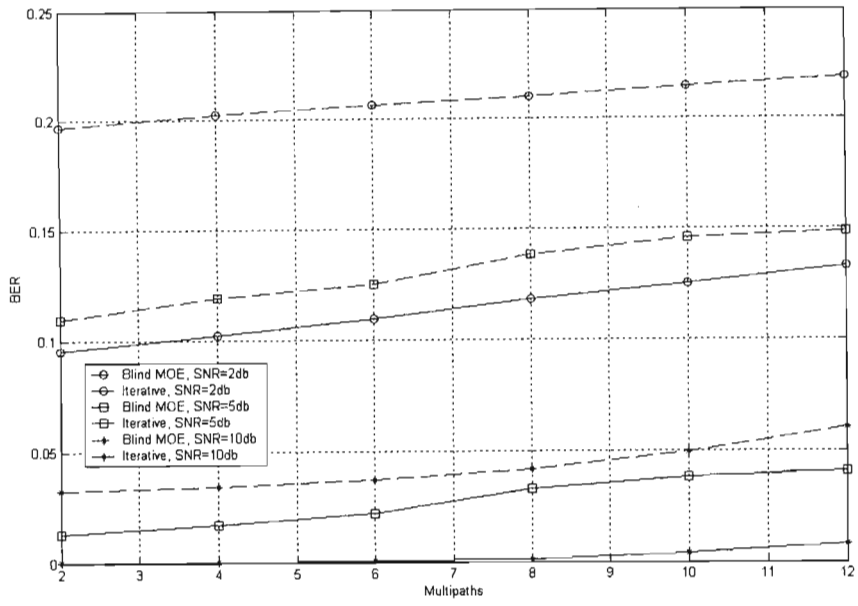
$$m = \sum_{k=1}^{K} \sum_{j=1}^{Lp} \alpha_j d\ (t - \tau_j) \tag{6.1}$$

Where L_p is the number of multipaths and k is the number of users, α_j is the attenuation relative to the parent signal, and τ_j is the time delay relative to the parent signal.

The performance of the iterative detector will again converge to that of the blind detector, for the same reasons as above.



6.10 (a)



(b)

Figure 6.10: The effect of multipaths for (a) 5 users, and (b) 10 users

6.3 Conclusion

The simulated and analytical results for the proposed blind iterative detector were presented in this chapter, and the performance of the proposed detector was compared to the performance of the blind MOE detector. From this comparison, it can be seen that the proposed detector has superior performance to that of the blind MOE receiver. It can also be seen that the analytical performance closely matches that of the simulated performance, and the proposed detector converges in three iterations. A system employing a rate-1/3 convolutional code shows an improved performance over a system with a rate-1/2 convolutional code, as predicted in Chapter 4, and an increase in processing gain results in a lower BER, as predicted in Chapter 3. There was a decrease in performance when the number of interfering users was increased, the number of multipaths was increased, or the power of the interfering users, relative to the desired user, was increased. This behaviour is supported by the theory presented in Chapter 3.

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Spread spectrum communication was first conceptualised in 1949, and direct sequence spread spectrum was proposed in 1950. The developments were initially restricted to military and navigation systems in the 1970s, until the cellular application of spread spectrum was realised in 1978. In 1986 the optimal multiuser detector was proposed by Verdu, and since a number of sub-optimal multiuser detectors, such as the MMSE and decorrelating detector, have been proposed. The first commercial venture using CDMA was the IS-95 standard (1993), and since then there has been commercialisation of wideband CDMA systems. CDMA cellular systems have recently been introduced in South Africa, and whilst the 3G technology has not been fully established, the superior data rates offered give it an advantage over the existing cellular and fixed line telecommunication systems.

The variant of CDMA we are concerned with is DS-CDMA (the other two being FH-CDMA and TH-CDMA). Its basic principle of operation is that the message bits for each user are spread with a unique pseudo-random sequence, which is nearly orthogonal to the spreading sequences of the other users operating on the channel. At the receiver, a chip-matched filter is used to de-spread the signal with the same sequence that it was spread with. As the signal components due to noise and the other users are not spread with that sequence, they are cancelled out. This method does not suppress interference entirely; as the spreading codes are not completely orthogonal there is still some MAI. As can be seen from Figure 3.2, the performance decreases as the number of users increases as there are more interfering components. Figures 3.4 and 3.5 show that the performance is degraded by multipaths and the near-far effect, respectively. Multiuser detectors attempt to mitigate these effects, a number of which have been proposed in literature. The most common are the MMSE, decorrelating and blind detectors, each having their own advantages and disadvantages. These detectors were simulated and their performances were compared in Figure 3.7. As can be seen, the performance of these three detectors is very close. The blind MOE detector has the distinct advantage of requiring less knowledge than the decorrelating or MMSE detectors.

Error coding is an integral part of any communication system. Errors do occur in CDMA systems, and the FEC coding has the ability to correct some, if not all, of these errors. The error coding schemes that are important to the aims of this report are convolutional codes and turbo codes. The iterative CDMA detectors require the iterative decoding that is employed in the decoding of turbo codes. The iterative CDMA system can be seen as a turbo code-like structure,

in that the convolutional code forms the "outer code," whilst the CDMA system acts as the "inner code." A number of iterative multiuser detectors have been proposed in literature, and selected detectors were described and compared in Section 5.1.3. These detectors, however, are primarily based on IC CDMA receivers, which require information about the channel, interfering users and the desired user. The MAP algorithm is the preferred decoder for these detectors.

The main goal of this research was to develop a blind iterative detector with error correcting capabilities. The proposed detector was introduced in Section 5.3.2. The advantage of this detector is that it has the iterative structure, but as it is blind, only requires knowledge of the desired user's spreading sequence. The initial detection is performed by a blind MOE receiver, which provides the first soft-inputs and estimates of the channel in order to calculate the priors. A SOVA decoder replaces the commonly used MAP decoder. Whilst the SOVA decoder has a slightly inferior performance to the MAP decoder, as can be seen in Figure 4.11, it is less computationally complex.

The proposed blind iterative structure was simulated using a (2, 1, 1) convolutional code. The simulated and analytical plots showing the performance of the proposed blind iterative detector were compared to the performance plots of the blind MOE detector. The superior performance of the proposed detector was apparent. The performance characteristics of the proposed blind iterative detector is summarised in Section 7.1.

In the future this research may be adapted using different blind algorithms, FEC coding, or iterative decoding techniques. There may be a possibility of using a completely blind receiver that requires no knowledge of the desired or interfering users, or the channel, similar to that of the EM approach proposed in [51] and [52]. A modification may be done on the iterative blind adaptive algorithm proposed in [50] to include error coding in the implementation of the algorithm. The convolutional code in the transmitter may be replaced with a block code, as iterative decoding of block codes has been proven to be possible [59]. The general structure of the blind iterative decoders proposed in this report and in [54] may be altered to allow for the implementation of other blind algorithms.

7.1 Research Conclusions

Blind multiuser detection using the minimum output energy criterion is an efficient way to decode the desired user's signal, as it requires the minimum amount of information to do so, namely the desired user's spreading sequence. Iterative multiuser detection has the advantage of having integrated error coding, the CDMA detector and error decoder operating in a similar fashion to turbo codes, in that they pass information to each other.

An iterative blind multiuser detector with FEC coding was proposed in this dissertation as a solution to the need for a multiuser detector than can mitigate the effects of the near-far problem and multipaths. It has the obvious advantage of both the detectors mentioned above: it requires the minimum knowledge and has integrated error coding. The blind MOE detector provides an estimation of the channel parameters and the initial soft decision input so that the priors may be calculated. The priors that have been calculated are fed into a SOVA decoder, the output of which is returned to the prior calculation to be updated for the next iteration.

The proposed detector was simulated, and the main performance features of the simulation results are summarised below:

- It exhibits superior performance compared to the blind MOE detector under various channel conditions.
- The iterative detector converges in three iterations.
- Using rate-1/3 convolutional codes result in improved performance over systems using rate-1/2 convolutional codes.
- The employment of rate-1/3 codes generates results showing a performance similar to that of the system proposed in [54].
- Rate-1/3 codes should be used when the interference is known to be great or when computation time is not an issue.
- Increase in pseudo-code length results in better performance
- The performance of the proposed system decreases with an increase in the channel population or the number of multipaths.
- The performance also decreases with an increase in the interfering user's signal strength in relation to the desired user's signal strength.

REFERENCES

[1] S. Verdu, *Multiuser Detection*, Cambridge, U.K.: Cambridge Univ. Press, 1998.

[2] William C. Y. Lee, "Overview of Cellular CDMA," *IEEE Trans. Vehic. Tech.*, Vol. 40, No. 2, May 1991.

[3] Herbert Taub and Donald L. Schilling, Principles of Communication Systems, 2nd ed., New Delhi, India: Tata McGraw-Hill, 1991.

[4] Timo O. Korhonen, Digital Communications lecture presentations, 2003, Helsinki University of Technology Communication Laboratory, available at: http://www.comlab.hut.fi/opetus/227/index.html

[5] Tero Ojanpera and Ramjee Prasad, *Wideband CDMA for Third Generation Mobile Communications*, Artech House, Boston, 1998.

[6] Aditya Dua, "Multiuser Detection in DS-CDMA – A Review," available at http://www.stanford.edu/~dua/.

[7] Aditya Dua, "MPOE Based Multiuser Detection for DS-CDMA Communications," Master of Technology thesis, Dept. Electron. Eng., Indian Institute of Technology, Mumbai, India, 2002.

[8] Ruxandra Lupas and Sergio Verdú, "Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channels," *IEEE Trans. Info. Theory*, Vol. 35, No. 1, pp.123-136, Jan. 1989.

[9] Upamanyu Madhow, "Multiuser Detection – An Overview and a new Result," invited paper, *International Workshop on Independent Component Analysis and Blind Signal Separation*, Helsinki, Finland, June 2000

[10] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple access channels," *IEEE Trans. Inform. Theory*, Vol. IT-32, No.1, pp.85-96, Jan. 1986.

[11] Alexandra Duel-Hallen, Jack Holtzman, and Zoran Zvonar, "Multiuser Detection for CDMA Systems," *IEEE Personal Communications*, pp. 46-58, April 1995.

[12] A.M. Bravo, J. Monera-LLorca, and V.K. Bhargava, "Implementation of the Decorrelating Receiver for Asynchronous DS-CDMA Systems over Mulipath Fading Channels," *Wireless Personal Communications* 15, pp.79-95, 2000.

[13] Shimon Moshavi, "Multiuser Detection for DS-CDMA Communications," *IEEE Communications*, pp.124-136, Oct. 1996.

[14] Shimon Moshavi, Emmanuel G. Kanterakis, and Donald L. Schilling, "A New Multiuser Detection Scheme for DS-CDMA Systems," *IEEE Milcom '95*, pp.518-522, 1995.

[15] HV Poor and S Verdú, "Probability of Error in MMSE Multiuser Detection," *IEEE Trans. Inform. Theory*, Vol. 43, No. 3, pp.858-871, May 1997.

[16] Ali F. Almutairi *et al*, "Perfomance of MMSE Receiver Based on CDMA System with Higher Order Modulation Formats in a Fading Channel," *Wireless Personal Communications* 25, 2003, pp.117-136.

[17] Sergio Verdú, "Adaptive Multiuser Detection," Spread Spectrum Techniques and Applications, 1994 IEEE ISSSTA '94, IEEE 3rd International Symposium on, 4-6 July 1994, vol. 1, pp.43-56.

[18] Zhou Jian and Yue Guangxin, "Blind Adaptive Multiuser Detection in Multipath Code Division Multiple Access Channels," Communication Technology Proceeding, 2000 WCC-ICCT 2000, International Conference on, Vol. 2, 21-25 Aug. 2000, pp.1325-1329.

[19] Michael Honig, Upamanyu Madhow, and Sergio Verdú, "Blind Adaptive Multiuser Detection," *IEEE Trans. Inform. Theory*, vol. 41, No. 4, pp. 944-960, July 1995.

[20] John G. Proakis, *Digital Communications*, 4th edition, McGraw Hill, New York, 2001.

[21] R. Michael Buejrer, Neiyer S. Correal-Mendoza, and Brian D. Woerner, "A Simulation Comparison of Multiuser Receivers for Cellular CDMA," *IEEE Trans. Vehic. Tech.*, Vol. 49, No. 4, pp.1065-1085, July 2000.

[22] A Burr, "Turbo-codes: the ultimate error control codes?" *Electronics and Communications Engineering Journal*, pp. 155-165, Aug. 2001.

[23] William E. Ryan, "A Turbo Code Tutorial," New Mexico State University, Las Cruces, available at http://www.ece.arizona.edu/~Wryan/.

[24] Jason P. Woodard and Lajos Hanzo, "Comparative Study of Turbo Decoding Techniques: An Overview," *IEEE Trans. Vehic. Tech.*, vol. 49, no. 6, pp. 2208-2233, Nov. 2000.

[25] S. Benedetto, R. Garello, and G. Montorsi, "A Search for Good Convolutional Codes to be Used in the Construction of Turbo Codes," *IEEE Trans. Commun.*, vol. 46, no. 9, pp. 1101-1105, September 1998.

[26] B. Sklar, "A Primer on Turbo Code Concepts," *IEEE Communications*, pp. 94-102, Dec. 1997.

[27] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, March 1974.

[28] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proc. ICC '93*, pp. 1064-1070, May 1993.

[29] P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," *Proc. ICC '95*, pp. 1009-1013.

[30] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," *Proc. GlobeCom 1989*, pp. 1680-1686.

[31] J. Hagenauer, "Source-Controlled Channel Decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2449-2457, Sept 1995.

[32] Steven S. Pietrobon, "Implementation and Performance of a Turbo/MAP Decoder," Int'l. J. Satellite Communications, vol. 15, Jan-Feb 1998, pp. 23-46.

[33] S. Dolinar, D. Divsalar, and F. Pollara. Code Performance as a Function of Block Size, NASA. Available online: http://tmo.jpl.nasa.gov/tmo/progress_report/42-133/title.htm.

[34] H.R. Sadjadpour, N.J.A. Sloane, M. Salehi, and G. Nebe, "Interleaver Design for Turbo Codes," IEEE JSAC, vol. 19, no. 5, pp. 831-837, May 2001.

[35] A.S. Barbulescu and S.S. Pietrobon, "Interleaver Design for Turbo Codes," *IEE Electron. Lett.*, pp. 2107-2108, Dec. 1994.

[36] Joseph Thomas and Evaggelos Geraniotis, "Joint Iterative MMSE Multiuser Detection and Narrowband Jammer Suppression in Coded DS-CDMA Channels," *Proc. IEEE Milcom '99.*

[37] Francesco Saverio Ostuni, Mohammad Reza Nakhai, and Hamid Aghvami, "Iterative Multi-user MMSE Receiver for Space-Time Trellis Coded CDMA Systems over Frequency Selective Channels," *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, 7-10 Sept. 2003, Vol. 2, pp 01- 43.

[38] Mohamed L. Ammari, Huu Tuê Huynh, and Paul Fortier, "Joint iterative Turbo decoding and estimation of correlated Rayleigh fading channel," *Proc. 7th Canadian Workshop on Information Theory*, June 3 – 6, 2001.

[39] Ralf R. Müller and Giuseppe Caire, "Efficient Implementation of Iterative Multiuser Decoding," *Proc. ISIT 2002*, Lausanne, Switzerland, June 30- July 5, 2002.

[40] Markku J. Junti, Behnaam Aazhang, and Jorna O. Lilleberg, "Iterative Implementation of Linear Multiuser Detection for Dynamic Asynchronous CDMA Systems," *IEEE Trans. Commun.*, vol. 46, No. 4, pp. 503-508, April 1998.

[41] Joachim Wehinger, Ralf Müller, Maja Lonëar, and Cristoph Mecklenbräuker, "Performance of Iterative CDMA Receivers with Channel Estimation in Multipath Environments," *Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, 2002, vol. 2, pp. 1439 – 1443.

[42] Hesham El Gamal and Evaggelos Geraniotis, "Iterative Multiuser Detection for Coded CDMA Signals in AWGN and Fading Channels," *IEEE JSAC*, vol. 18, No. 1, Jan. 2000.

[43] Alex Grant and Christian Schlegel, "Convergence of Linear Interference Cancellation Multiuser Receivers," *IEEE Trans. Commun.*, vol. 49 No. 10, pp. 1824-1834, Oct. 2001.

[44] Slavica Marinkovic, Branka Vucetic, and Akihisa Ushirokawa, "Space-Time Iterative and Multistage Receiver Structures for CDMA Mobile Communication Systems" *IEEE JSAC*, vol. 19, No. 8, August 2001.

[45] Christian Schlegel and Zhenning Shi, "Performance and Complexity of CDMA Iterative Multiuser Detection," *Proc. ITW2003*, Paris, March 31- April 4, 2003.

[46] P. B. Darwood, P. D. Alexander, K. Wacker, and I. J. Oppermann, "Iterative Multi-User Detection and Channel Estimation for CDMA with non-Binary Modulation," *Proc. Globecom 2001*, San Antonio, 2001.

[47] Paul D. Alexander, Mark C. Reed, John A. Asenstorfer, and Christian B. Schlegel "Iterative Multiuser Interference Reduction: Turbo CDMA," *IEEE Trans. Commun.*, vol. 47, No. 7, pp. 1008-1014, July 1999.

[48] Mark C. Reed, Christian B. Schlegel, Paul D. Alexander, and John. A. Asenstorfer, "Iterative Multiuser Detection for CDMA with FEC: Near-Single-User Performance," *IEEE Trans. Commun.*, vol. 46, no. 12, pp. 1693-1699, Dec 1998.

[49] Bo Xia and William E. Ryan, "Near-Optimal Convolutionally Coded Asynchronous CDMA with Iterative Multiuser Detection/Decoding," *Proc. 2002 IEEE International Conference on Communications*, Vol. 3, pp. 1521 -1525, April/May 2002.

[50] R.T. Derryberry, Tan F. Wong and James S. Lehnert, "An Iterative Blind Adaptive Receiver for DS-SSMA Systems," *Proc. IEEE Milcom '98.*

[51] E. Khan and D. Slock, "Iterative Blind Demodulation of Synchronous CDMA," *Proc. PIMRC*, Lisbon, Portugal, Sept. 2002.

[52] Y. Yao and H.V. Poor, "Eavesdropping in the synchronous CDMA Channel: An EM Based Approach," *IEEE Trans. on Signal Processing*, vol. 49, no. 8, pp. 1748-1756, Aug. 2001.

[53] Ejaz Khan, and Dirk Slock, "Blind Iterative Receiver for Multiuser Space-Time Coding Systems," *Proc. 6th Baiona Workshop on Signal Processing in Communications*, 8-10 Sept. 2003.

[54] Teng Joon Lim, Tao Zhu, and Mehul Motani, "Blind Iterative Decision-Feedback Multiuser Detection of FEC-Coded CDMA Signals," *IEEE Commun. Lett.*, vol. 5, no. 11, pp.459-461, Nov. 2001.

[55] G. Woodward and M.L. Honig, "Performance of Adaptive Iterative Multiuser Parallel Decision Feedback with Different Code Rates," in *Proc. IEEE Int. Conf. Comms. (ICC) 2001*, Helsinki, Finland, June 2001

[56] Z. Shi and C. Schlegel, "Joint Iterative Decoding of Serially Concatenated Error Control Coded CDMA," *IEEE JSAC*, vol. 19, no. 8, pp. 1646-1653, Aug. 2001.

[57] Alex J. Grant and Paul D. Alexander, "Convergence Analysis for Iterative Multiuser Decoding," *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications,* London, UK.

[58] J. Whitehead and F. Takawira, "Performance Analysis of MMSE ST-MUD of DS-CDMA with a Correlated Vector Channel Model," *SAIEE Transactions*, vol. 94, no. 4, pp. 9-14, Dec 2003.

[59] Joachim Hagenauer, Elke Offer, and Lutz Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. 42, No. 2, pp. 429-445, March 1996.

PUBLICATIONS FROM THE THESIS

- B van Niekerk and S.H. Mneney, "Blind Iterative MUD with Error Coding," presented at *IEEE Africon 2004*, Gabarone, Botswana, 2004.
- Brett van Niekerk and Stanley H. Mneney, "Blind Iterative MUD for FEC Coded CDMA," submitted to *SAIEE Elektron*, 2005
- Brett van Niekerk and Stanley H. Mneney, "Blind Iterative MUD for FEC Coded CDMA," submitted to *IEE Electronics Letters*, 2005
- Brett van Niekerk and Stanley H. Mneney, "Blind Iterative MUD for Error Coded CDMA," submitted to *SAIEE Transactions*, 2005