Iterative Multi-User Turbo-Code Receiver for DS-CDMA

by

Dwayne Stienstra

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical Engineering

Waterloo, Ontario, Canada, 2000

©Dwayne Stienstra 2000
I hereby declare that I am the sole author of this research report.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this research report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.
The University of Waterloo requires the signatures of all persons using or photocopying this research report. Please sign below, and give address and date.
Abstract

The topic of interference cancellation in a coded Code Division Multiple Access (CDMA) system has been the focus of much recent research. Earlier works have studied methods for jointly decoding all the users resulting in most having an exponential complexity for the corresponding receiver. In this thesis, a number of different iterative decoding methods are proposed for the multi-user interference cancellation in a CDMA system where Turbo Codes are utilized for forward error correction. In the proposed decoding schemes, the individual users are decoded separately with the operation of iterative interference cancellation being mixed with the iterative decoding of Turbo Codes.

These iterative decoders result in only a marginal increase in the overall complexity as compared to a conventional single user receiver utilizing Turbo Codes for forward error correction. Numerical performance results are presented showing that in the cases of practical interest for CDMA applications, the multiple access interference can be essentially removed at a reasonable level of complexity. These iterative decoders achieve a similar to better performance with a reduction in complexity as compared to similar previously known research work.
Acknowledgements

I would like to acknowledge my supervisor Dr. A.K. Khandani for his guidance throughout my research. Thank you for your patience and insight about the use of Turbo Codes in CDMA applications.

I would like to thank NSERC, CITO, and the University of Waterloo for their financial support.

I would also like to thank my colleagues at the University of Waterloo for their help and support in my research. Most of all, a sincere thanks to my parents and family for their moral support and understanding.
# Contents

1 Introduction 1

2 Direct-Sequence CDMA System Model 11
   2.1 Introduction .................................................. 11
   2.2 System Description ........................................... 12
      2.2.1 Continuous-Time System Model ......................... 12
      2.2.2 Discrete-Time System Model ......................... 14
   2.3 Turbo Code Encoder ........................................ 17

3 Iterative Multi-User CDMA Receiver 21
   3.1 Introduction .................................................. 21
   3.2 Turbo Code Decoder ........................................... 24
   3.3 Iterating the CDMA Decoder ................................. 30
      3.3.1 Continuous Gaussian Approximation of MAI .......... 33
      3.3.2 Discrete Analysis of MAI ............................... 35
      3.3.3 Correlated Gaussian Noise Model ..................... 36
      3.3.4 Extending The Correlated Gaussian Noise Model .... 41
4 Numerical Results

4.1 Simulation Setup ........................................... 49
4.2 Continuous Gaussian Approximation of MAI BER Performance .. 52
4.3 Discrete Analysis of MAI BER Performance .......................... 54
4.4 Continuous Approximation Versus Discrete Analysis of MAI BER
    Performance .................................................. 57
4.5 Correlated Gaussian Noise Model BER Performance ............... 63
4.6 Extended Correlated Gaussian Noise Model BER Performance ... 64
4.7 Complexity Analysis ........................................... 66

5 Conclusions .................................................. 72

5.1 Future Research Directions ................................... 73

A Complexity Calculations ...................................... 75

A.1 Turbo Code Complexity ...................................... 76
A.2 Final Hard Decision Complexity ............................... 78
A.3 Metric Generator Complexity ................................ 78
A.4 Overall Decoder Complexity ................................ 88

Bibliography .................................................... 94
List of Tables

4.1 Relative Computational Complexity Per User For Continuous Gaussian Approximation and Discrete Analysis of MAI Receivers . . . . . . . . . 69
4.2 Relative Computational Complexity Per User For Continuous Gaussian Approximation of MAI, Correlated and Extended Correlated Gaussian Noise Model Receivers . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
## List of Figures

1.1 Digital Communication System ........................................ 1
1.2 Multiple Access Communication System ........................... 3

2.1 Continuous-Time CDMA System Model .............................. 13
2.2 Discrete-Time CDMA System Model .................................. 15
2.3 Rate 1/3 Turbo Code Encoder ....................................... 18

3.1 Iterative Multi-User Decoder ........................................ 23
3.2 Two-Code Turbo Code Decoder ...................................... 26

4.1 Continuous Approximation of MAI with $K = 5$ Users and Block Size $M = 192$ for Various Values of $\rho$ ..................... 51
4.2 Continuous Approximation of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$ .......................... 53
4.3 Continuous Approximation of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$ ......................... 53
4.4 Continuous Approximation of MAI with $K = 2$ Users, Block Size $M = 192$ and Chip Length $N = 7$ ......................... 55
4.5 Continuous Approximation of MAI with $K = 2$ Users, Block Size $M = 600$ and Chip Length $N = 7$ .......................... 55
4.6 Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$ .......................... 56
4.7 Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$ .......................... 56
4.8 Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 192$ and Chip Length $N = 7$ .......................... 58
4.9 Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 600$ and Chip Length $N = 7$ .......................... 58
4.10 Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$ .......................... 59
4.11 Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$ .......................... 59
4.12 Continuous Approximation Versus Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 192$ and Chip Length $N = 7$ .......................... 61
4.13 Continuous Approximation Versus Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 600$ and Chip Length $N = 7$ .......................... 61
4.14 Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$ .......................... 62
4.15 Hard Versus Soft Decision of Noise with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$ .......................... 65
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.16</td>
<td>Hard Versus Soft Decision of Noise with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$</td>
<td>65</td>
</tr>
<tr>
<td>4.17</td>
<td>Extended Correlated Gaussian Noise Model with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$</td>
<td>67</td>
</tr>
<tr>
<td>4.18</td>
<td>Extended Correlated Gaussian Noise Model with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$</td>
<td>67</td>
</tr>
<tr>
<td>4.19</td>
<td>Extended Correlated Gaussian Noise Model with $K = 7$ Users, Block Size $M = 192$ and Chip Length $N = 7$</td>
<td>68</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

A digital wireless communication system consists of many basic elements as shown in Figure 1.1. These elements include:

- Source encoder, which converts the information from the source into digital form, removing redundancy. The source is restored back again at the destination through the source decoder.

- Channel encoder, which adds controlled redundancy to the information sequence in such a way that the channel decoder can overcome the effects of

Figure 1.1: Digital Communication System
CHAPTER 1. INTRODUCTION

the noise and interference caused by transmission over the noisy channel.

- Modulator and demodulator, which transforms the digital signal into an analog waveform that can be transmitted efficiently over the channel and converted back into a digital format at the receiver side.

- Communication channel is the physical medium over which the information is transmitted to the destination where the transmitted signal is usually corrupted in a random manner by additive thermal noise, man-made noise, and atmospheric noise [5].

The channel encoder can add the controlled redundancy using one of two known strategies: Forward Error Correction (FEC) and Automatic Repeat Request (ARQ). Error correcting codes are used in FEC encoders in order that the receiver can correct errors caused by the channel. ARQ encoders utilize an error detecting code in order that a retransmission can be requested when the receiver detects an error.

In 1993, Claude Berrou et al. [1] introduced a class of FEC parallel concatenated convolutional codes known as Turbo Codes that produce results close to the Shannon limit\(^1\). The Turbo Code technique combines the concepts of iterative decoding\(^2\), soft-in/soft-out decoding, and non-uniform random interleaving. The corresponding decoding algorithm is a modification of the BCJR [2] algorithm applied to the decoding of Recursive Systematic Convolutional (RSC) codes. Berrou

\(^{1}\)Claude Shannon presented theoretical limits of the performance of a communication system in his 1948 paper [6].

\(^{2}\)An iterative decoder is a suboptimal algorithm that provides extremely good performance while requiring only a modest level of complexity. An iterative decoder operates on the received data in a repetitive manner utilizing feedback to form estimates of the transmitted data.
CHAPTER 1. INTRODUCTION

Figure 1.2: Multiple Access Communication System

et al. reported a Bit Error Rate (BER) of $10^{-5}$ for a bit Signal to Noise Ratio (SNR) of 0.7 dB. This performance was a large improvement over other coding techniques at the time and has lead to much research in the area of Turbo Codes. Turbo Codes will be used as a means of FEC in this thesis since they perform much better than other channel coding techniques. The use of other channel coding techniques is also possible for the proposed decoders of this thesis.

The channel in a communication system is typically used by more than a single user. A simple multiple access communication system is illustrated in Figure 1.2, where $K$ independent users share the same additive noise channel. The receiver in this system can be a common receiver for the decoding of all users or several receivers for each of the $K$ users. Multiple access communication systems can be of two types: multipoint-to-point (uplink) or point-to-multipoint (downlink). The mobile cellular communication system is an example of a multiple access communication system.

There are various ways in which multiple users can access a shared channel to transmit information in an efficient and reliable manner. The three main methods include Frequency-Division Multiple Access (FDMA), Time-Division Multiple Access (TDMA), and Code-Division Multiple Access (CDMA), of which there are
many variations and combinations possible. In FDMA systems, each user is assigned a different carrier frequency so the resulting spectrums do not overlap. For TDMA systems, time is partitioned into slots with each user being assigned to a particular slot in which to transmit information. Both FDMA and TDMA systems can be rather inefficient when various users transmit bursty information, as in voice communications, since a certain percentage of available carrier frequencies or time slots will not be carrying information [5].

A CDMA\textsuperscript{3} system incorporates spread spectrum technology to allow each user access to the entire spectrum, making better use of the limited bandwidth and time resources available to the entire system. Spread spectrum systems have been used in military applications for many years because of their ability to reject interference caused by multipath and jamming (intentional or not) and have only within the last decade been implemented in commercial communication systems. These systems have been found attractive because of the characteristics of potential capacity increases, universal frequency reuse, anti-multipath capabilities, soft capacity, and soft handoff [15]. As such, there has been much recent research in the area of CDMA systems. This is the multiple access technique used in the communication system of this thesis.

In a CDMA system\textsuperscript{4}, each user is assigned a unique noiselike spreading signal that is used to spread the user information sequence over the assigned frequency

\textsuperscript{3}CDMA systems are grouped into a number of different categories depending on the spreading technique, such as direct-sequence CDMA (DS-CDMA) and frequency-hopped CDMA (FH-CDMA). Most of the recent research has focused on DS-CDMA. This thesis will only focus on DS-CDMA and, thus, CDMA will be used in place of DS-CDMA henceforth.

\textsuperscript{4}Only a brief description of a CDMA system is presented here. For a more detailed description, see references [5] and [7].
band, using more bandwidth than what is necessary to transmit the information. The spread information signal is then transmitted over the shared channel.

The spreading signals for the users in the system can be designed to be orthogonal resulting in a simple receiver. However, in practice there will be bandwidth or complexity constraints as well as synchronization constraints among users that prevent the spreading signal set from being orthogonal. Thus, the spreading signals must be carefully selected so that their cross-correlations are fairly low as compared to the energy of the spreading signals, in order that the interfering effect of other users in the system is reduced. The spread information signal looks like random interference to all the other users, which can be partially suppressed at the receiver while decoding each user signal. Because of this Multiple Access Interference (MAI), the performance of CDMA systems is mainly governed by the number of users in the system where increasing the number of users increases the amount of interference, resulting in a degraded system performance.

The performance of a CDMA system is also limited by the near-far effect that occurs because non-orthogonal code signals are used. The near-far effect occurs when some distant user signals are partially or completely masked out at the receiver by users that are close to the base station. In the absence of power control, the nearer users are received at a much higher power level than the distant users resulting in severe degradation of the weak user’s signal. CDMA systems are very sensitive to the near-far effect, with the effect of MAI on system performance being even more pronounced. Thus, CDMA systems typically employ fast and accurate power control systems in order that all received user power levels differ only by a
CHAPTER 1. INTRODUCTION

few dB at the base station.

The received signal in a CDMA system is a superposition of the individual transmitted information signals. If orthogonal spreading signals are used, a bank of single-user detectors, consisting of a filter matched to the specific spreading signals and a threshold detector\(^5\), would achieve optimum detection [5]. This is the simplest receiver for the CDMA system and can also be used when the spreading signals are not orthogonal. This receiver is typically referred to as the conventional CDMA receiver in the literature. It decodes each user separately and does not take into account the existence of MAI, treating the other users as additive interference or noise. The performance of the conventional receiver is reasonable for non-orthogonal code signals when the number of users is small but deteriorates rapidly as the number of users increases.

Since the spread information signals from the different users are correlated when non-orthogonal code signals are used, the conventional receiver is sub-optimal. For non-orthogonal code signals, CDMA performance can be greatly enhanced by jointly decoding\(^6\) all the user signals instead of decoding them separately as done in the conventional detector. The task of these detectors, referred to as multi-user detectors, is to reliably decode the information signal for each user. Most multi-user detectors typically are used in conjunction with the conventional receiver.

Sergio Verdú first introduced the optimal uncoded multi-user detector in his Ph.D thesis [8] and discussed it further in his subsequent papers [9, 10]. His deriva-

\(^5\)A bank of correlators in place of the matched filters can equivalently be utilized to implement this receiver.

\(^6\)The joint decoding process is also referred to as multi-user detection or interference cancellation in the literature.
tion of the optimum multi-user detector was based on a maximum likelihood se-
quenue detection formulation for asynchronous users transmitting over an additive
white Gaussian noise (AWGN) channel. This derivation can be easily extended for
the simpler case of synchronous users. The complexity of the optimal multi-user
detector increases exponentially with the number of users resulting in its not being
feasible for implementation in a practical communication system.

Since the main drawback of the optimal multi-user detector is complexity, most
of the recent research works have addressed the problem of simplifying multi-user
detection for implementation in a practical communication system. There is a wide-
range of possible performance versus complexity tradeoffs possible for multi-user
detection. Most of these sub-optimal multi-user receivers try to reduce or remove
the effect of the MAI on each user.

Early works on multi-user detection for uncoded synchronous systems are re-
ported in [16, 17, 18]. Some linear complexity sub-optimal receivers that have been
proposed include the decorrelating detector [19, 20], the minimum mean-square-
error sequence detector [21], and the projection receiver [22]. Several other sub-
optimal multi-user detectors that utilize feedback to reduce the effect of the MAI in
the received signal include multistage detectors [23, 24], decision-feedback detectors
[25, 26], and successive or parallel interference cancellation [27, 28]. Duel-Hallen
et al. [15] and the references within provide a further description of most of these
multi-user receivers.

\footnote{In synchronous CDMA systems, the bit and chip time intervals are aligned at the receiver while for asynchronous CDMA systems the bit and or chip time intervals may not be aligned.}
CHAPTER 1. INTRODUCTION

Most of the proposed sub-optimal multi-user detectors in the literature are designed with no consideration to channel coding being present in the system. Only recently has there been a focus on developing multi-user detectors in combination with various channel coding methods. The multi-user decoders proposed in [12-14, 29-38] incorporate an FEC code in the design of the decoders, resulting in improved BER performances at lower SNR values.

Gaillorenzi and Wilson [31] extended the optimal uncoded decoder of Verdú [8] to incorporate convolutional coding for FEC. They showed that this decoder could achieve single-user performance for reasonable cross-correlation values typical in CDMA systems. However, the complexity of this decoder made it unfeasible in a practical communication system as the complexity is exponentially dependent on the number of users and the number of states of each encoder.

Moher [34, 35] proposed a sub-optimal multi-user decoder with an iterative structure similar to Turbo Codes utilizing convolutional codes for FEC. This iterative decoder achieved near single-user performance and is less complex than Giallorenzi and Wilson’s [31] optimal decoder, still being exponentially dependent on the number of users. Moher’s work focused mainly on highly correlated systems, like FDMA, and is not of interest to this thesis.

Reed et al. [12, 13, 14] also proposed an iterative multi-user decoder, similar to Moher, for synchronous CDMA systems. These decoders utilize both convolutional codes [12, 13] and Turbo Codes [14] for FEC. In both cases, near single-user performance was achieved. However, the complexity in both decoders is still exponential with respect to the number of users.
CHAPTER 1. INTRODUCTION

Alexander, Grant, and Reed [40] further extended their previous work [12, 13, 14] to the asynchronous CDMA system. Their proposed sub-optimal iterative decoder operates on a sequence of received symbols. It views the concatenation of direct-sequence spreading with the asynchronous multiple-access channel as a special form of convolutional code, implementing an interference cancellation operation as part of its decoding operation. Near single-user performance is obtained with this decoder.

This thesis proposes a number of different sub-optimal iterative multi-user decoders for CDMA, utilizing Turbo Codes for channel coding and decoding, over an AWGN channel. The iterative structure of these decoders is similar to the iterative structure utilized by both Moher [34, 35] and Reed et al. [12, 13, 14], where the joint detection is combined with channel decoding. The proposed decoders essentially use the individual Turbo Code decoders to perform the interference cancellation process while adding minimal complexity to the Turbo Codes. This is the main difference between the proposed decoders and those of both Moher [34, 35] and Reed et al. [12, 13, 14]. This difference yields complexity that is polynomial in the number of users, $K$, for the proposed schemes as well as a comparable or better performance in cases of practical interest.

All of the proposed multi-user decoders in this thesis utilize the same iterative structure, incorporating the iterative, soft in/soft out structure of Turbo Codes. The proposed decoders differ from each other by a probability distribution metric that is generated for use in each individual Turbo decoder. The difference between these decoders is a result of varying approximations for the probability distribution
of the MAI and the method employed for updating the conditional probability distributions in the system.

The thesis is organized as follows: Chapter 2 examines the CDMA system model used in the development of the proposed iterative multi-user decoders and also examines the Turbo Code encoder. Chapter 3 reviews the Turbo Code decoder used in the multi-user receiver and presents the derivation of the proposed iterative decoders. Chapter 4 describes the simulation setup and presents the BER performance and complexity results of the proposed schemes. Finally, Chapter 5 concludes with a summary of the results and suggests some future research directions.
Chapter 2

Direct-Sequence CDMA System Model

2.1 Introduction

The system model used throughout this thesis is that of the uplink of a CDMA communication system. This model is based on the same assumptions as used by Reed et al. [12, 13, 14] and Verdù [11]. These assumptions include perfect power control\(^1\), i.e., all user signals are received at the same power level, and that the channel is both chip and symbol synchronous\(^2\). This chapter examines how these assumptions affect this model. The Forward Error Correction (FEC) channel coding that is used for each user in the system is also discussed.

\(^1\)The acquisition and maintenance of each user’s power levels is beyond the scope of this thesis. See [39] and references within for more detailed information on power control.

\(^2\)Note that the generalization to an asynchronous channel is straightforward if the MAI term is treated properly.
2.2 System Description

2.2.1 Continuous-Time System Model

The basic continuous-time $K$ user CDMA channel model used in this thesis, shown in Figure 2.1, consists of a sum of modulated synchronous signature waveforms transmitted over an additive white Gaussian noise (AWGN) channel. Each user transmits $M$ source bits. The chip and symbol synchronous, continuous-time CDMA signal for one source bit at the receiver is represented as:\(^3\)

$$r(t) = \sum_{k=1}^{K} A_k d_k s_k(t) + \sigma n(t)$$  \hspace{1cm} (2.1)

over the time interval $t \in [0, T]$, where $T$ is the inverse of the data rate utilized by each user. In (2.1), the term $s_k(t)$ represents the spreading signal utilized by user $k$ normalized to unit energy, $\int_0^T s_k^2(t) dt = 1$, and is assumed to be zero outside the interval $[0, T]$ so there is no intersymbol interference. Other components of (2.1) include the received amplitude $A_k$ of the $k^{th}$ user signal, where $d_k \in \{-1, +1\}$ is the $k^{th}$ user coded bit, and $n(t)$ is white Gaussian noise with unit power spectral density added to the received signal, where the one-sided spectral level is $N_0 = 2\sigma^2$. Since the assumption of perfect power control has been made, the received amplitude $A_k$ can be normalized to 1 resulting in unit energy, $A_k^2 = 1$, for all $K$ users\(^4\).

---

\(^3\)Throughout this thesis, the notations follow the standard: scalars are lower case, vectors are highlighted and lower case, and matrices are highlighted and upper case.

\(^4\)This thesis does not consider fading because of the perfect power control assumption. In general, if the amplitude of fading is known, which is the assumption in most studies on Turbo Codes, one would only need to incorporate its effect in the noise variance.
The performance of various decoding schemes will depend not only on the signal
to noise ratio, but also on the amount of correlation between the spreading signals
of the $K$ users. The cross-correlation of the spreading signals of user $i$ and user $j$
is defined by the equation:

$$\rho_{ij}(t) = \int_{0}^{T} s_i(t)s_j(t)dt. \quad (2.2)$$

The cross-correlation values may be time-varying if random spreading signals are
utilized. From the Cauchy-Schwartz inequality \[5\] and the normalization $\int_{0}^{T} s_k^2(t)dt = 1$, it is known that\footnote{5} $|\rho_{ij}(t)| \leq 1$. The cross-correlation values can be collected into a
cross-correlation matrix $H_t = \{\rho_{ij}(t)\}$, which has diagonal elements equal to 1 and
is symmetric nonnegative definite\footnote{6}.

\footnote{5}{In this case, the symbol $|\cdot|$ represents the absolute value operation. For the remainder of this
thesis it is used for the determinant operation.}

\footnote{6}{The cross-correlation matrix $H_t$ is positive definite if and only if the spreading signals $s_k(t)$,
$k \in \{1, \ldots, K\}$, are linearly independent.}
2.2.2 Discrete-Time System Model

The multi-user detector commonly has a front-end processing block that converts the continuous-time signal $r(t)$ into a discrete-time signal $r_t$. The front-end of the receiver for this thesis uses $K$ coherently matched filters\(^7\), each matched to its corresponding user spreading signal $s_k(t)$, the output of which is sampled at rate $T$.

No information relevant to the decoding of the received signal is lost by the conversion to discrete-time signals by the matched filtering process [11]. Thus, the sampled output of the matched filters represents sufficient statistics for the decoding operation. The discrete-time CDMA system model will be used in this thesis to reduce the complexity of the equations.

The block diagram of the equivalent discrete-time CDMA system is shown in Figure 2.2. The $K$ users each transmit $M$ source bits $b_r^{(k)} \in \{0, 1\}$ per block of data, where $k \in \{1, \ldots, K\}$ specifies the user and $r \in \{1, \ldots, M\}$ is the uncoded time index of the block of data. Information is transmitted on a block by block basis due to the decoding process and will be discussed in the next section. The source bits $b_r^{(k)}$ are assumed to be 0 or 1 with probability of 1/2 and are independent of the other $K - 1$ users. The uncoded bits $b_r^{(k)}$ are coded into $L = M/R$ symbols $d_t^{(k)} \in \{-1, +1\}$ per block, where $k \in \{1, \ldots, K\}$, $t \in \{1, \ldots, L\}$ is the coded time index\(^8\) of the block of data, and $R$ is the coding rate. The coded symbols are

---

\(^7\)Note that a RAKE receiver with a few modifications could also be used for the front-end of this receiver.

\(^8\)Note that the time variable $t$ is used in both the continuous-time and discrete-time models. For the remainder of this thesis $t$ is used as a discrete variable.
spread by the spreading signals before transmission over the channel. The discrete-time spreading signal utilized by user $k$ at symbol interval $t$ consists of $N$ chips, normalized to unit energy, and is denoted by $s_t^{(k)} \in \left\{ -\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}} \right\}$. In CDMA systems, a spreading signal is divided into chips, each of equal duration smaller than the bit duration. The values of these chips comprise the discrete-time spreading signal. The decoder is assumed to have full knowledge of the spreading signals of all the users.

The sampled output of the matched filter for the $k^{th}$ user can be expressed as

$$r^{(k)}_t = \int_0^T r(t)s_k(t)dt,$$  \hspace{1cm} (2.3)

9The continuous-time and discrete-time spreading signals are related by the equation $s_k(t) = \sum_{n=1}^{N} a_n^{(k)} p(t-nT_c)$, where $t \in [0, T]$, $a_n^{(k)} \in \left\{ -\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}} \right\}$ is the $n^{th}$ chip value, $T_c$ is the chip interval of shorter duration than $T$, and $p(t)$ is a rectangular pulse of duration $T_c$. The resulting discrete-time spreading signal is the length $N$ sequence of chip values $\{a_1^{(k)}, \ldots, a_N^{(k)}\}$. See reference [5] for further details.
which can be expanded using (2.1) and (2.2) as

\[ r_t^{(k)} = d_t^{(k)} + \sum_{i=1, i \neq k}^{K} d_t^{(i)} p_t^{(i k)} + n_t^{(k)}, \]  

(2.4)

where

\[ n_t^{(k)} = \sigma \int_{0}^{T} n(t) s_k(t) dt \]  

(2.5)

is a Gaussian random process with zero mean and variance \( \sigma^2 = N_0/2 \) [11].

The matched filter outputs of the \( K \) users can be combined and expressed in vector form as

\[ \mathbf{r}_t = \mathbf{H}_t \mathbf{d}_t + \mathbf{n}_t, \]  

(2.6)

where\(^\text{10}\) \( \mathbf{r}_t = \left( r_t^{(1)}, \ldots, r_t^{(K)} \right)^\top \) is the received vector, \( \mathbf{d}_t = \left( d_t^{(1)}, \ldots, d_t^{(K)} \right)^\top \) is the coded data vector, and \( \mathbf{H}_t \) is the \( K \times K \) discrete-time cross-correlation matrix of the spreading signals. The noise samples \( \mathbf{n}_t = \left( n_t^{(1)}, \ldots, n_t^{(K)} \right)^\top \), as a result of matched filtering, are correlated with autocorrelation matrix \( E \left\{ \mathbf{n}_t \mathbf{n}_t^\top \right\} = \mathbf{H}_t \sigma^2 \). The \( E\{\} \) operation represents the expectation operator for the remainder of this thesis.

As mentioned in Chapter 1, the FEC channel coding method used in this thesis is Turbo Codes. The Turbo Code encoder is examined in the next section while the Turbo Code decoder is examined in the next chapter.

\(^{10}\)The symbol \( (\cdot)^\top \) represents the transposition operation and is used throughout this thesis.


2.3 Turbo Code Encoder

A two-code Turbo Code encoder\(^{11}\) consists of a parallel concatenation of two binary RSC encoders. Figure 2.3 illustrates a two-code turbo encoder with two identical RSC component codes, each with \(P = 4\) memory elements. Each RSC encoder will go through a sequence of states, i.e.,

\[
S_0^M = \{S_0, S_1, \ldots, S_M\}. \tag{2.7}
\]

Each \(S_r\) fully describes the state of the encoder and the previous transitions. The encoder is assumed to start and end in the all zero state, i.e., \(S_0 = S_M = 0\). The RSC encoder with generator polynomials \(g_1 = 37\)\(_8\) and \(g_2 = 21\)\(_8\) (denoted in octal base) is used for both RSC encoders in this thesis.

The first RSC encoder uses the \(k^{th}\) user’s source bit stream \(\{b_r^{(k)}\}\) to generate two output bit streams:

1. \(\{X_r^{(k)}\}\), the systematic bits generated by replicating the source bits \(b_r^{(k)}\);
2. \(\{Y_{1,r}^{(k)}\}\), the parity bits generated by the first encoder.

The encoding is performed on the \(k^{th}\) user’s source bit stream block by block, where the size of the block is \(M\). Note that turbo codes are essentially equivalent to block codes with the trellis of the encoder terminated at the end of each block. For the second RSC encoder, the \(k^{th}\) user’s source bit stream \(\{b_r^{(k)}\}\) is passed through a pseudo-random interleaver to form the second RSC encoder input sequence \(\{b_{\nu}^{(k)}\}\),

\(^{11}\)Note that the proposed decoders in this thesis can be generalized to any size Turbo Code.
$\nu \in \{1, \ldots, M\}$. Here, the $b_{r}^{(k)}$ and $b_{\nu}^{(k)}$ are related by $\nu = \text{INT}(r)$, meaning that the $r^{th}$ bit in the original data sequence has been moved to the $\nu^{th}$ position in the interleaved sequence. The second binary RSC encoder uses this pseudo-random interleaved source bit sequence to generate the parity bit sequence $\{Y_{2,\nu}^{(k)}\}$.

The parallel concatenation of the two RSC encoders produces a rate $R = 1/3$ code. The resulting coded sequence of length $L$ is:

$$\left\{ (X_{r}^{(k)}, Y_{1,r}^{(k)}, Y_{2,\nu}^{(k)}) \right\} = \left\{ (X_{1}^{(k)}, Y_{1,1}^{(k)}, Y_{2,\text{INT}(1)}^{(k)}) , \ldots , (X_{M}^{(k)}, Y_{1,M}^{(k)}, Y_{2,\text{INT}(M)}^{(k)}) \right\}. \quad (2.8)$$

This bit stream is Binary Phase-Shift Key (BPSK) modulated to produce the length
CHAPTER 2. DIRECT-SEQUENCE CDMA SYSTEM MODEL

$L$ transmitted sequence:

$$\left\{ d_t^{(k)} \right\} = \left\{ \left( 2X_r^{(k)} - 1, 2Y_{1,r}^{(k)} - 1, 2Y_{2,\nu}^{(k)} - 1 \right) \right\}, \quad t \in \{1, \ldots, L\}. \tag{2.9}$$

Each coded symbol $d_t^{(k)}$ is spread by the $k^{th}$ users spreading signal $s_t^{(k)}$ and transmitted over the AWGN channel.

The matched filter output of the CDMA receiver for all $K$ users at time $t$ is given in (2.6) as $r_t = H_t d_t + n_t$, where $r_t = \left( r_t^{(1)}, \ldots, r_t^{(K)} \right)$. The noise $n_t = \left( n_t^{(1)}, \ldots, n_t^{(K)} \right)$ is correlated among the $K$ users, with the autocorrelation matrix $E \left\{ n_t n_t^\top \right\} = H_t \sigma^2$, but is still bit-wise independent for each user, i.e., $E \left\{ n_t n_v^\top \right\} = 0$ for $t \neq v$. For a synchronous system, the whole received sequence for the data block at the output of the bank of matched filters is represented by $r^\top = \left( r_1^\top, \ldots, r_L^\top \right)$.

The received sequence of length $L$ for user $k$ can be further expressed in terms of the systematic and parity bits as $\left\{ r_t^{(k)} \right\} = \left\{ (x_r^{(k)}, y_{1,r}^{(k)}, y_{2,\nu}^{(k)}) \right\}, \quad t \in \{1, \ldots, L\}$, $r, \nu \in \{1, \ldots, M\}$, where from (2.4):

$$x_r^{(k)} = \left( 2X_r^{(k)} - 1 \right) + \sum_{i=1 \atop i \neq k}^K \left( 2X_r^{(i)} - 1 \right) \rho_r^{(ik)} + \zeta_r^{(k)} \tag{2.10}$$

$$y_{1,r}^{(k)} = \left( 2Y_{1,r}^{(k)} - 1 \right) + \sum_{i=1 \atop i \neq k}^K \left( 2Y_{1,r}^{(i)} - 1 \right) \rho_{1,r}^{(ik)} + \eta_r^{(k)} \tag{2.11}$$

$$y_{2,\nu}^{(k)} = \left( 2Y_{2,\nu}^{(k)} - 1 \right) + \sum_{i=1 \atop i \neq k}^K \left( 2Y_{2,\nu}^{(i)} - 1 \right) \rho_{2,\nu}^{(ik)} + \xi_{\nu}^{(k)} \tag{2.12}$$

$\zeta_r^{(k)}$, $\eta_r^{(k)}$, and $\xi_r^{(k)}$ are the correlated noise terms and $\rho_r^{(ik)}$, $\rho_{1,r}^{(ik)}$, and $\rho_{2,\nu}^{(ik)}$ are the correlation values corresponding to the systematic and parity bits. The received
variables $r_t^{(k)}$ are bit-wise independent of each other but are correlated to the other users respective systematic and parity bits as a result of matched filtering. These received values are used by the decoder to produce an estimate of each users original bit $b_r^{(k)}$. 
Chapter 3

Iterative Multi-User CDMA Receiver

3.1 Introduction

In CDMA systems, the received signal, at the output of the matched filters, consists of the original signal, channel noise, and Multiple Access Interference (MAI), as shown in (2.4). MAI is the result of not using orthogonal spreading signals. This interference can have a tremendous effect on the performance of the receiver when the number of users increases. Thus, the design of the CDMA decoder is very important for the overall performance of the system.

The conventional, and simplest, decoder for CDMA systems consists only of individual matched filters, matched to each user’s spreading signal. It decodes each user’s signal as if it were the only one present in the system. This decoder leads to severe performance degradation as the number of users in the system increase due
to the increased MAI. This poor performance can be reduced somewhat by the use of powerful channel coding.

The performance of the receiver has been found to improve greatly by utilizing multi-user decoders that compensate for the MAI. A multi-user decoder jointly estimates the transmitted symbols of all the users instead of estimating them independently, where the decision criteria is based on minimizing the sequence or bit error probability. As mentioned in Chapter 1, the optimum “uncoded” multi-user detector, based on maximum likelihood sequence detection for asynchronous Gaussian channels, was introduced in 1984 by Sergio Verdú [8]. However, the computational complexity of the optimum decoder is very high, generally growing exponentially with the number of users. As a result, much research has focused on finding good sub-optimal multi-user receivers. Several iterative decoders have also been examined. The sub-optimal iterative multi-user CDMA decoders that are proposed in this thesis utilize an iterative structure similar to Turbo Code decoders, where the iterations of the decoders are used to cancel the effect of the MAI.

The iterative multi-user CDMA decoder is illustrated in Figure 3.1. As can be seen from the figure, soft information is passed during the iterations, in a similar manner as Turbo Codes. The decoder is composed of a metric generator and $K$ individual FEC decoders. The metric generator uses the outputs of the matched filters, $r_t = (r_t^{(1)}, \ldots, r_t^{(K)})^\top$, from the front end of the receiver to generate a metric suitable for use in the $K$ individual Turbo Code decoders. Each Turbo Code decoder uses this metric to produce conditional probabilities, which will be subsequently used as the $a$-priori input to the metric generator for use in the next
CHAPTER 3. ITERATIVE MULTI-USER CDMA RECEIVER

The iterative process continues in this manner until further iterations yield little or no significant improvement.

The main problem of implementing the iterative multi-user CDMA decoder is that of generating the correct probability information in the metric generator for use in the FEC decoders. From the output of the matched filters, \( \mathbf{r}_t = \mathbf{H}\mathbf{d}_t + \mathbf{n}_t \), we know that the conditional probability of \( \mathbf{r}_t \) is a multivariate Gaussian distribution \( p(\mathbf{r}_t|\mathbf{d}_t) \) [5]. However, the Turbo Code decoders take advantage of the marginal conditional probabilities corresponding to a single-user instead of using the whole multivariate Gaussian distribution, in order to reduce complexity. The key point to the proposed iterative algorithm is to find a proper method to update the corresponding marginal probability distribution, \( p(\mathbf{r}^{(k)}_t|\mathbf{d}^{(k)}_t) \), from iteration to iteration. This thesis explores different methods to extract and update these marginal probabilities.
3.2 Turbo Code Decoder

As the Turbo Code encoder consists of two RSC encoders, the Turbo Code decoder consists of two RSC decoders for each user. These decoders are used iteratively to improve the Log-Likelihood Ratio (LLR) produced for each input bit. To calculate the LLR of each bit, the A Posteriori Probability (APP) of each encoded bit is needed. The BCJR [2] algorithm estimates the APP values of the states and transitions of a Markov source observed through a discrete memoryless channel. Berrou et al. [1] applied a variation of the BCJR algorithm for the decoding of RSC encoders. Robertson [3] and Jung [4] have made further improvements to reduce the complexity and memory storage of the algorithm.

This section will review the basic notation of the modified BCJR algorithm and the modifications proposed in [3]. A detailed version of the derivation of the BCJR algorithm can be found in [2] and will not be discussed here. For notational simplicity, the parameter indicating the user will be dropped throughout this section as an identical algorithm will be used in the Turbo Code decoder of each user. Also, the derivation of the algorithm in this section is for a single-user. Modifications of the Turbo Code decoder for the iterative CDMA multi-user decoder will be discussed in the next section.

An RSC decoder operates on the received sequence block $R_1^M = (R_1, \ldots, R_M)$, where $R_r = (x_r, y_r)$. The values $x_r$ and $y_r$ are the RSC encoded parameters for data bit $r$ that have been sent over the noisy channel and received at the output of the matched filter bank. The RSC decoder uses $R_1^M$ to calculate the LLR value
for each bit. The LLR is given by the equation:

\[
\Lambda(b_r) = \ln \left( \frac{P_r \{ b_r = 1 | R_1^M \}}{P_r \{ b_r = 0 | R_1^M \}} \right),
\]

(3.1)

where \( P_r \{ b_r = i | R_1^M \} \) is the APP of the input bit \( b_r, i \in \{0,1\} \). The LLR values are generated by forward and backward recursions through the trellis of the RSC encoder.

The forward recursive intermediate variable is denoted by \( \alpha_r(m) \) and takes into account the effect of past observations. The backward recursive intermediate variable, on the other hand, takes into account future observations and is denoted by \( \beta_r(m) \). These recursive variables are calculated by the following equations:

\[
\alpha_r(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_i(R_r, m', m) \alpha_{r-1}(m')}{\sum_{m'} \sum_{i=0}^1 \gamma_i(R_r, m', m) \alpha_{r-1}(m')} \quad \text{(3.2)}
\]

\[
\beta_r(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_i(R_{r+1}, m', m) \beta_{r+1}(m')}{\sum_{m'} \sum_{i=0}^1 \gamma_i(R_{r+1}, m', m) \alpha_r(m')} \quad \text{(3.3)}
\]

The derivations of these two equations can be found in [3]. In these equations, \( \gamma_i(R_r, m', m) = P_r \{ R_r, S_r = m, b_r = i | S_r-1 = m' \} \) is the branch transition probability function which is a function of the trellis structure, the APP values of \( b_r \), and the channel statistics. The LLR is calculated from \( \alpha, \beta, \) and \( \gamma \) through the equation:

\[
\Lambda(b_r) = \ln \left( \frac{\sum_m \sum_{m'} \gamma_1(R_r, m', m) \alpha_{r-1}(m') \beta_r(m)}{\sum_m \sum_{m'} \gamma_0(R_r, m', m) \alpha_{r-1}(m') \beta_r(m)} \right).
\]

(3.4)

The Turbo Code decoder suggested by Robertson [3] uses this algorithm in an iterative manner with two RSC decoders and is shown in Figure 3.2 for the \( g^{th} \) iter-
CHAPTER 3. ITERATIVE MULTI-USER CDMA RECEIVER

Figure 3.2: Two-Code Turbo Code Decoder

ation. Each of the RSC encoders use the modified BCJR algorithm to calculate the LLR values of each source bit. The first component decoder receives the sequence \( \{R^1_r\} = \{R^1_1, \ldots, R^1_M\} \), where \( R^1_r = (x_r, y^1_{1,r}) \), while the second component decoder receives the sequence \( \{R^2_\nu\} = \{R^2_{\text{INT}(1)}, \ldots, R^2_{\text{INT}(M)}\} \), where \( R^2_\nu = (x_\nu, y^2_{2,\nu}) \). The second RSC decoder does not start decoding until the first RSC decoder has completed its operations. The sequence \( \{x_\nu\} \) is the interleaved version of the received systematic information sequence \( \{x_r\} \). The variables \( x_r, y^1_{1,r}, \) and \( y^2_{2,\nu} \) were previously given in (2.10), (2.11), and (2.12) respectively for user \( k \).

The information that each decoder receives from the other decoder is used to update the branch transition probabilities \( \gamma_i^{(g)}(R^1_r, m', m) \) and \( \gamma_i^{(g)}(R^2_\nu, m', m) \) for the \( g^{th} \) decoding iteration. In the following, the equations for the first RSC decoder are derived while the equations for the second decoder are similarly obtained, with the substitutions of \( b_\nu \) for \( b_r \) and \( y^2_{2,\nu} \) for \( y^1_{1,r} \). The branch transition probabilities for the first encoder are expressed as:

\[
\gamma_i^{(g)}(R^1_r, m', m) = P_r \{ b_r = i, R^1_r, S_r = m | S_{r-1} = m' \}
\]
\[ p\left(R_r^1|b_r = i, S_r = m, S_{r-1} = m'\right) \]
\[ P_r\left\{b_r = i|S_r = m, S_{r-1} = m'\right\} \]
\[ P_r^{(g)}\{S_r = m|S_{r-1} = m'\}. \quad (3.5) \]

Since \(x_r\) and \(y_{1,r}\) are uncorrelated random variables, the first term in (3.5) can be expanded as:

\[ p\left(R_r^1|b_r = i, S_r = m, S_{r-1} = m'\right) = p(x_r|b_r = i, S_r = m, S_{r-1} = m') \]
\[ p(y_{1,r}|b_r = i, S_r = m, S_{r-1} = m'). \quad (3.6) \]

However, the systematic bits \(x_r\) are independent of the trellis structure and thus,

\[ p(x_r|b_r = i, S_r = m, S_{r-1} = m') = p(x_r|b_r = i). \quad (3.7) \]

The second term in (3.5), \(P_r\{b_r = i|S_r = m, S_{r-1} = m'\}\) is determined by the trellis structure. \(P_r\{b_r = i|S_r = m, S_{r-1} = m'\} = 1\) only if there is a path between the previous state \(m'\) and the current state \(m\) for an input \(b_r = i\), otherwise \(P_r\{b_r = i|S_r = m, S_{r-1} = m'\} = 0\). When there is a path between states \(m'\) and \(m\), the state transition probability \(P_r^{(g)}\{S_r = m|S_{r-1} = m'\}\) of the trellis is defined as:

\[ P_r^{(g)}\{S_r = m|S_{r-1} = m'\} = P_r^{(g)}\{b_r = i\} = \text{a priori probability of bit } b_r. \quad (3.8) \]
CHAPTER 3. ITERATIVE MULTI-USER CDMA RECEIVER

Only the state transition probabilities need to be updated in the iterative decoding process.

The branch transition probability can be rewritten as:

\[ \gamma_i^{(g)}(R^1_r, m', m) = P_r^{(g)} \{ b_r = i \} \gamma'_i(R^1_r, m', m), \]  

(3.9)

where

\[ \gamma'_i(R^1_r, m', m) = p(x_r|b_r = i) p(y_{1,r}|b_r = i, S_r = m, S_{r-1} = m') \]

\[ P_r \{ b_r = i | S_r = m, S_{r-1} = m' \}. \]  

(3.10)

The components \( \gamma_i^{(g)}(R^1_r, m', m) \) and \( \gamma'_i(R^2_r, m', m) \) are constant and are not updated in the Turbo decoding iterations. Thus, these values are calculated after receiving the values \( x_r, y_{1,r}, \) and \( y_{2,\nu}. \)

Both RSC decoders separately produce LLRs of the received source bits, with each decoder using information from the other decoder. For the \( g^{th} \) decoding iteration, the LLR of the first decoder can be factored as:

\[ \Lambda_1^{(g)}(b_r) = \ln \left( \frac{p(x_r|b_r = 1)}{p(x_r|b_r = 0)} \right) + \ln \left( \frac{P_r^{(g)} \{ b_r = 1 \}}{P_r^{(g)} \{ b_r = 0 \}} \right) + \]

\[ \ln \left( \frac{\sum_{m} \sum_{m'} \gamma''_{1} (y_{1,r}, m', m) \alpha_{r-1}^{(g)} (m') \beta_{r}^{(g)} (m)}{\sum_{m} \sum_{m'} \gamma''_{0} (y_{1,r}, m', m) \alpha_{r-1}^{(g)} (m') \beta_{r}^{(g)} (m)} \right) \]

\[ = I_r + L_2^{(g-1)}(b_r) + L_1^{(g)}(b_r), \]  

(3.11)
where

\[
\gamma''_{1}(y_{1,r}, m', m) = p(y_{1,r}|b_{r} = i, S_{r} = m, S_{r-1} = m')
\]

\[
P_{r}\{b_{r} = i|S_{r} = m, S_{r-1} = m'\}. \tag{3.12}
\]

Only the extrinsic information \(L_{1}^{(g)}(b_{r})\) is passed to the second RSC decoder. The extrinsic information \(L_{2}^{(g-1)}(b_{r})\), which was originally generated by the second decoder, and the intrinsic\(^1\) information \(I_{r}\) do not need to be passed to the second decoder. For the first decoder, the state transition probabilities defined in (3.8) for iteration \(g\) are generated using the extrinsic information \(L_{2}^{(g-1)}(b_{r})\) through the equation:

\[
P_{r}^{(g)}\{b_{r} = i\} = \frac{\exp \{iL_{2}^{(g-1)}(b_{r})\}}{1 + \exp \{L_{2}^{(g-1)}(b_{r})\}}. \tag{3.13}
\]

The second RSC decoder uses the received sequence \(\{R_{2}^{\nu}\}\) and the extrinsic information \(L_{1}^{(g)}(b_{r})\) to produce the LLR \(\Lambda_{2}^{(g)}(b_{\nu})\). The extrinsic information \(L_{1}^{(g)}(b_{r})\) is interleaved to \(L_{1}^{(g)}(b_{\nu})\) before use in the second decoder. This extrinsic information is used to update the state transition probabilities of \(\gamma_{i}^{(g)}(R_{2}^{\nu}, m', m)\) through the equation:

\[
P_{r}^{(g)}\{b_{\nu} = i\} = \frac{\exp \{iL_{1}^{(g)}(b_{\nu})\}}{1 + \exp \{L_{1}^{(g)}(b_{\nu})\}}. \tag{3.14}
\]

The LLR \(\Lambda_{2}^{(g)}(b_{\nu})\) for the second decoder can also be factored as:

\[
\Lambda_{2}^{(g)}(b_{\nu}) = I_{\nu} + L_{1}^{(g)}(b_{\nu}) + L_{2}^{(g)}(b_{\nu}). \tag{3.15}
\]

\(^{1}\)Intrinsic information is provided by the systematic bits \(x_{r}\), while the extrinsic information is provided by the encoded, or parity, bits \(y_{r}\). This distinction was introduced by Berrou et al. [1].
The extrinsic information $L^{(g)}_1(b_r)$ and $L^{(g)}_2(b_\nu)$ are also considered to be Log-Likelihood Ratios. The extrinsic information $L^{(g)}_2(b_\nu)$ is de-interleaved and passed to the first decoder for the next iteration.

The extrinsic information $L^{(0)}_2(b_r)$ is set to zero for the first iteration. The forward and backward variables are initialized, prior to the start of decoding, for both RSC decoders to:

1. $\alpha_0(0) = 1$ and $\alpha_0(m) = 0$ for $m \neq 0$;
2. $\beta_M(0) = 1$ and $\beta_M(m) = 0$ for $m \neq 0$.

These initial values result from the initial and final states being assumed to be zero, i.e., $S_0 = \mathbf{0}$ and $S_M = \mathbf{0}$.

Only after a set number of iterations have been completed are the LLR values $\Lambda^{(g)}_2(b_\nu)$ compared to get an estimate of the bit $b_r$ transmitted. The LLR values $\Lambda^{(g)}_2(b_\nu)$ are first de-interleaved to $\Lambda^{(g)}_2(b_r)$. A final hard-decision is made by comparing $\Lambda^{(g)}_2(b_r)$ to a threshold of zero. Thus,

$$\hat{b}_r = \begin{cases} 1 & : \Lambda^{(g)}_2(b_r) > 0 \\ 0 & : \Lambda^{(g)}_2(b_r) \leq 0 \end{cases} \quad (3.16)$$

and the magnitude of the LLR value indicates the reliability of this decision.

### 3.3 Iterating the CDMA Decoder

For the iterative multi-user CDMA receiver, each Turbo Code decoder is modified to produce both uncoded and coded bit probabilities $P_r \left\{ b_r^{(k)} = b | r^{(k)} \right\}$, $b \in \{0, 1\}$,
and \( P_r \{ d_t^{(k)} = d | r_t^{(k)} \} \), \( d \in \{-1, 1\} \), respectively. The coded bit probabilities are combined from the output of the first and second RSC decoders of the Turbo Code decoders since each decoder uses a different coded sequence as a result of interleaving. The uncoded bit probabilities are extracted from the output of the second decoder.

As previously mentioned, the conditional probability of the output of the matched filters, \( r_t \), is a multivariate Gaussian distribution \( p (r_t | d_t) \). However, each individual Turbo Code decoder requires the probability distribution \( p \left( r_t^{(k)} | d_t^{(k)} \right) \), as in (3.7), as the input metric. The coded bit probabilities are used by the metric generator in the updating of \( p \left( r_t^{(k)} | d_t^{(k)} \right) \).

The metric generator creates the metric \( p \left( r_t^{(k)} | d_t^{(k)} \right) \) by assigning the output of the Turbo Code decoders from iteration \( i \) as the \textit{a-priori} input probability for the \( i + 1 \) iteration in order to form the proper probability distribution. This process is similar to Turbo Code decoding discussed in the previous section where the output probability of the first decoder is used as the \textit{a-priori} information for the second decoder and vice versa. For the initial iteration, the metric generator assumes the bit probabilities \( P_r \left\{ d_t^{(k)} = -1 \right\} \) and \( P_r \left\{ d_t^{(k)} = 1 \right\} \) for the \( k^{th} \) user are equally likely and equal to \( 1/2 \). For subsequent iterations, the metric generator assigns the coded bit probabilities created by the individual Turbo Code decoders from the previous iteration to the individual \textit{a-priori} bit probabilities of the next iteration through the equation:

\[
P_r \left\{ d_t^{(k)} = d \right\} = P_r \left\{ d_t^{(k)} = d | r_t^{(k)} \right\}, \quad d \in \{-1, +1\}.
\] (3.17)
After the required iterations have been completed, the systematic bit probabilities \( P_r \{ b_r^{(k)} = b | r^{(k)} \} \), \( b \in \{0, 1\} \), generated by the \( K \) Turbo Code decoders are used to make the final hard-decision to calculate \( \hat{b}_r \), as in (3.16). Note that during the multi-user iterations, each Turbo Code decoder continues to use the output of one component decoder as the \textit{a-priori} probability for the other decoder as in normal Turbo decoding.

The problem for the metric generator is the selection of the probability distribution \( p \left( r^{(k)}_t | d^{(k)}_t \right) \) for the \( k^{th} \) Turbo Code decoder. The following sections discuss different methods to compute this probability distribution. To reduce the complexity of the derivations for the probability metric, the cross-correlation\(^2\) term \( \rho_{t}^{(ij)} \) in (2.2) is assumed to be constant and equal for all \( i \neq j, i, j \in \{1, \ldots, K\} \) and time invariant for all \( t \), i.e.,

\[
\rho_{t}^{(ij)} = \begin{cases} 
\rho & \forall i \neq j, t \\
1 & i = j.
\end{cases}
\]  

(3.18)

As a result of this assumption, the discrete cross-correlation matrix \( H_t \) becomes a time invariant matrix, represented as \( H \), which has diagonals equal to one and the non-diagonal elements equal to \( \rho \). The generalization to the time-variant matrix \( H_t \) is straightforward for all the proposed decoders except the discrete analysis of MAI decoder, since only the probability distribution of \( p \left( r^{(k)}_t | d^{(k)}_t \right) \) is required in the iterative process. The discrete analysis decoder is derived by utilizing the binomial distributed nature of the MAI when \( \rho \) is constant and is difficult to extend when the cross-correlations are time-varying. As will be shown in Chapter 4, the

\(^2\)The computation of the cross-correlation values is performed by the receiver, which is assumed to have full knowledge of all the user’s spreading signals. See [11] for further details.
continuous Gaussian approximation of the MAI decoder performs similarly with lower complexity than the discrete analysis of the MAI decoder. Thus, the discrete analysis of the MAI decoder need not be extended for time-varying cross-correlation values since the continuous Gaussian approximation of the MAI decoder would be utilized instead.

3.3.1 Continuous Gaussian Approximation of MAI

From (2.4), the output of the $k^{th}$ matched filter consists of the coded data bit $d_t^{(k)}$, the MAI from other users $\rho \sum_{i=1, i \neq k}^{K} d_t^{(i)}$, and a Gaussian noise term $n_t^{(k)}$ with zero mean and variance $\sigma^2$ that is correlated with autocorrelation matrix $E\{n_t n_t^\top\} = H\sigma^2$. The MAI term consists of $K - 1$ discrete binary random variables. From the central limit theorem\(^3\), as the number of users $K$ increases, the distribution of the MAI term approaches that of a Gaussian. In one class of algorithms proposed in this thesis, the MAI term is approximated by a Gaussian random variable.

Each individual user binary probability distribution consists of the probabilities $P_r\{d_t^{(k)} = 1\}$ and $P_r\{d_t^{(k)} = -1\}$, $k \in \{1, \ldots, K\}$. The mean and variance of these binary distributions for the $k^{th}$ user, $k \in \{1, \ldots, K\}$ are calculated as:

\[
\begin{align*}
    m_t^{(k)} &= E\{d_t^{(k)}\} = \sum_{d \in \{-1, 1\}} d P_r\{d_t^{(k)} = d\} \\
    &= P_r\{d_t^{(k)} = 1\} - P_r\{d_t^{(k)} = -1\} \\
    \psi_t^2(k) &= E\{d_t^2(k)\} - E\{d_t^{(k)}\}^2 \tag{3.19}
\end{align*}
\]

\(^3\)The central limit theorem states that the sum of $n$ statistically independent random variables approach a Gaussian distribution as $n \to \infty$. \hfill \square
\[
\begin{align*}
&= \sum_{d \in \{-1, 1\}} d^2 P_r \{ d_t^{(k)} = d \} - \left[ \sum_{d \in \{-1, 1\}} d P_r \{ d_t^{(k)} = d \} \right]^2 \\
&= P_r \{ d_t^{(k)} = 1 \} + P_r \{ d_t^{(k)} = -1 \} - \left[ P_r \{ d_t^{(k)} = 1 \} - P_r \{ d_t^{(k)} = -1 \} \right]^2 \\
&= 1 - \left[ P_r \{ d_t^{(k)} = 1 \} - P_r \{ d_t^{(k)} = -1 \} \right]^2.
\end{align*}
\]

(3.20)

The addition of these binary distributions results in the MAI term \( \rho \sum_{i=1}^{K} d_t^{(i)} \) for the \( k^{th} \) user having a Gaussian distribution with mean and variance:

\[
\begin{align*}
\mu_{t, \text{MAI}}^{(k)} &= \rho \sum_{i=1 \atop i \neq k}^{K} \left[ P_r \{ d_t^{(i)} = 1 \} - P_r \{ d_t^{(i)} = -1 \} \right] = \rho \sum_{i=1 \atop i \neq k}^{K} m_t^{(i)} \quad (3.21) \\
\sigma_{t, \text{MAI}}^2^{(k)} &= \rho^2 \sum_{i=1 \atop i \neq k}^{K} \left( 1 - \left[ P_r \{ d_t^{(i)} = 1 \} - P_r \{ d_t^{(i)} = -1 \} \right]^2 \right) = \rho^2 \sum_{i=1 \atop i \neq k}^{K} v_t^{2(k)}. \quad (3.22)
\end{align*}
\]

The probability distribution \( p \left( r_t^{(k)} | d_t^{(k)} = d \right) \) for the \( k^{th} \) user, \( k \in \{1, \ldots, K\} \) and \( d \in \{-1, 1\} \), is derived as a Gaussian random variable with mean and variance:

\[
\begin{align*}
\mu_t^{(k)} &= d + \mu_{t, \text{MAI}}^{(k)} \quad (3.23) \\
\sigma_t^2^{(k)} &= \sigma^2 + \sigma_{t, \text{MAI}}^2 \quad (3.24)
\end{align*}
\]

In the derivation of this probability distribution, the correlation of the noise between all users at time \( t \) is disregarded to simplify the analysis. The correlation of the noise will be taken into account in the metrics described in sections 3.3.3 and 3.3.4.
3.3.2 Discrete Analysis of MAI

The continuous Gaussian approximation of the MAI becomes accurate when the number of users becomes large. Thus, a different approach is needed to determine the probability distribution of the MAI when the number of users is small.

The MAI term $\rho \sum_{i=1, i \neq k}^{K} d_t^{(i)}$ for user $k$ is a discrete quantity that can only take on $K$ different values since each coded bit $d_t^{(i)}$ can only take on values $\{-1, 1\}$. As a result, the MAI term can be represented as a random variable with a binomial distribution. This probability distribution is obtained for user $k$ by convolving the other $K - 1$ sets of coded bit probabilities to produce the length $K$ discrete probability sequence:

$$\{P_{t,m}^{(k)}\} = \left\{ P_r \left\{ \sum_{i=1, i \neq k}^{K} d_t^{(i)} = m \right\} \right\} = \left\{ P_r \left\{ d_t^{(1)} = -1 \right\}, P_r \left\{ d_t^{(1)} = 1 \right\} \right\} * \ldots * \left\{ P_r \left\{ d_t^{(K-1)} = -1 \right\}, P_r \left\{ d_t^{(K-1)} = 1 \right\} \right\} * \ldots * \left\{ P_r \left\{ d_t^{(K)} = -1 \right\}, P_r \left\{ d_t^{(K)} = 1 \right\} \right\}, \quad (3.25)$$

where $m \in \{- (K - 1), - (K - 3), \ldots, (K - 3), (K - 1)\}$ and "*" is the discrete convolution operator.

---

4It is well known that if independent random variables are added, the corresponding probability distributions will convolve.

5Discrete convolution is defined as $c_l = \sum_{j=-\infty}^{\infty} a_{l-j} b_j = a_l * b_l$ for $l \in (-\infty, \infty)$, where $a_l$ and $b_l$ are discrete sequences.
CHAPTER 3. ITERATIVE MULTI-USER CDMA RECEIVER

The probability distribution of the metric is determined by convolving the distributions of the MAI term and the Gaussian noise term of zero mean and variance $\sigma^2$. The correlation of the noise is disregarded in the derivation of the metric’s probability distribution. The convolution of the two distributions results in the probability distribution of the metric being a sum of Gaussian random variables expressed as:

$$p\left(r_t^{(k)}|d_t^{(k)} = d\right) = \sum_m \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{(r_t^{(k)} - d - \rho m)^2}{2\sigma^2}\right] \times P_{t,m},$$

(3.26)

where $m \in \{-K + 1, -K + 3, \ldots, K - 3, K - 1\}$ and $d \in \{-1, 1\}$.

3.3.3 Correlated Gaussian Noise Model

The previous two sections described two different probability distributions for the input metric for the individual Turbo Code decoders based on improving the estimate of the distributions of the MAI. However, both of these probability distributions are derived by ignoring the correlation of the Gaussian noise among the users, both treating the noise as being independent among the users. In this section, the correlation among the noise samples is used to produce better estimates for the probability distribution of $p\left(r_t^{(k)}|d_t^{(k)}\right)$.

The correlated Gaussian noise terms $n_t^{(k)}$, $k \in \{1, \ldots, K\}$, possess information about each other. If these terms were known, the correlation between the users could be removed. However, the noise terms are not known exactly and only estimates of them can be obtained through the use of the coded bit probabilities at the
output of the Turbo Code decoders from the previous iteration. The metric generator uses these estimates to produce a more accurate description of the probability distribution of the noise based on the information known about the noise.

Upon generating an estimate of the noise samples, the metric generator calculates the conditional distribution \( p \left( n_i^{(k)} | n_i^{(1)}, \ldots, n_i^{(K)} \neq k, n_i^{(k)} \right) \) to replace the noise distribution \( p( n_i^{(k)} ) \) in the calculation of the metric \( p \left( r_i^{(k)} | d_i^{(k)} \right) \). The next section will examine an extension of this method where a non-conditional distribution is used to replace the noise distribution. The conditional distribution \( p \left( n_i^{(k)} | n_i^{(1)}, \ldots, n_i^{(K)} \neq k, n_i^{(k)} \right) \) utilizes the \( K - 1 \) other estimated noise values to reduce the effect of the noise for user \( k \). From Bayes’ rule [5], the conditional noise distribution can be calculated from:

\[
p \left( n_i^{(k)} | n_i^{(1)}, \ldots, n_i^{(K)} \neq k, n_i^{(k)} \right) = \frac{p \left( n_i^{(1)}, \ldots, n_i^{(k)}, \ldots, n_i^{(K)} \right)}{p \left( n_i^{(1)}, \ldots, n_i^{(K)} \neq k, n_i^{(k)} \right)}.
\]

From the previous chapter, the noise vector \( \mathbf{n}_t = (n_t^{(1)}, \ldots, n_t^{(K)})^\top \) in (2.6) is known to be a zero mean, correlated multivariate Gaussian random variable with correlation \( E \left\{ \mathbf{n}_t \mathbf{n}_t^\top \right\} = \mathbf{H} \sigma^2 \), where \( \mathbf{H} \) is the \( K \times K \) discrete-time cross-correlation matrix. Thus, the noise vector \( \mathbf{n}_t \) has the probability distribution:

\[
p \left( \mathbf{n}_t \right) = p \left( n_t^{(1)}, \ldots, n_t^{(k)}, \ldots, n_t^{(K)} \right)
= \frac{1}{(2\pi)^{K/2} |\mathbf{H}|^{1/2}} \exp \left[ -\frac{1}{2} \mathbf{n}_t^\top \mathbf{H}^{-1} \mathbf{n}_t \right].
\]  \[ (3.28) \]  

The probability distribution of \( p \left( n_t^{(1)}, \ldots, n_t^{(K)} \right) \) will also be a multivariate Gaussian distribution where \( \mathbf{H} \) is a \( (K - 1) \times (K - 1) \) matrix.

Expanding and simplifying (3.27), the conditional noise probability distribution
takes the form:

\[
p \left( n_t^{(k)} | n_t^{(1)}, \ldots, n_t^{(K)} \right) = \frac{1}{\sqrt{2\pi\tilde{\sigma}_t^{(k)}}} \exp \left[ -\frac{(n_t^{(k)} - \tilde{\mu}_t^{(k)})^2}{2\tilde{\sigma}_t^{(k)}^2} \right], \tag{3.29}
\]

which is a Gaussian random variable with mean and variance:

\[
\tilde{\mu}_t^{(k)} = \frac{\rho}{(K - 2)\rho + 1} \sum_{i=1, i\neq k}^{K} n_t^{(i)}, \tag{3.30}
\]

\[
\tilde{\sigma}_t^{2(k)} = \tilde{\sigma}^2 = \sigma^2 \left[ \frac{(1 - \rho)(1 + (K - 1)\rho)}{(1 + (K - 2)\rho)} \right], \tag{3.31}
\]

respectively. It is observed that the conditional noise variance \( \tilde{\sigma}^2 \) is constant and does not need to be updated during the multi-user iterations. However, the mean \( \tilde{\mu}_t^{(k)} \) depends on the estimates of the noise produced by the metric generator and is updated during the multi-user iterations.

Generating the updated noise probability function is done in combination with the updating of the probability distribution for the MAI term. The continuous Gaussian approximation of the MAI is used in the updating of the input metric for the Turbo Code decoders in this section. Since the coded bit probabilities are used directly in the generation of the noise distribution, reliable bit probabilities are needed before updating the noise distribution. Due to this reason, the distribution of the noise is updated after a few iterations of the multi-user decoder have been completed in which only the MAI distribution is updated. The probability distribution function of the noise is updated with the coded bit probabilities in two ways: (i) by making a hard decision on the coded bit probabilities, and (ii) by
making a soft decision on the coded bit probabilities. These two methods will be further discussed in the following.

The probability distribution of the metric \( p \left(r_t^{(k)} | d_t^{(k)} = d \right), d \in \{-1, 1\}, \) including the effect of updating the MAI and the noise distribution, is a Gaussian random variable with mean and variance:

\[
\mu_t^{(k)} = d + \mu_{t, MAI}^{(k)} + \hat{\mu}_t^{(k)} \\
= d + \rho \sum_{i=1 \atop i \neq k}^{K} \left[ P_r \left\{ d_t^{(i)} = 1 \right\} - P_r \left\{ d_t^{(i)} = -1 \right\} \right] \\
+ \frac{\rho}{(K-2)\rho + 1} \sum_{i=1 \atop i \neq k}^{K} n_t^{(i)} \\
\sigma_t^{2(k)} = \sigma_{t, MAI}^{2(k)} + \sigma^2 = \rho^2 \sum_{i=1 \atop i \neq k}^{K} \left( 1 - \left[ P_r \left\{ d_t^{(i)} = 1 \right\} - P_r \left\{ d_t^{(i)} = -1 \right\} \right] \right)^2 \\
+ \sigma^2 \left[ \frac{(1 - \rho)(1 + (K-1)\rho)}{(1 + (K-2)\rho)} \right],
\]

where \( \mu_{t, MAI}^{(k)} \) and \( \sigma_{t, MAI}^{2(k)} \) are the MAI Gaussian distribution’s mean and variance given in (3.21) and (3.22).

**Hard Decision on Coded Bit Probabilities**

For the correlated Gaussian noise model of this section, the metric generator makes a hard decision on the coded bit probabilities received from the individual Turbo Code decoders as:

\[
d_t^{(k)} = \begin{cases} 
-1 : & P_r \left\{ d_t^{(k)} = -1 \right\} \geq P_r \left\{ d_t^{(k)} = 1 \right\} \\
1 : & P_r \left\{ d_t^{(k)} = -1 \right\} < P_r \left\{ d_t^{(k)} = 1 \right\} .
\end{cases}
\]
The estimated coded bits are used to produce an estimate of the noise for user $k$ at time $t$ through the equation:

$$n_t^{(k)} = r_t^{(k)} - d_t^{(k)} - \rho \sum_{i=1 \atop i \neq k}^{K} d_t^{(i)}.$$  \hfill (3.35)

These estimates of the noise are used in the calculation of the mean (3.32) of the probability metric $p \left( r_t^{(k)} | d_t^{(k)} \right)$.

**Soft Decision on Coded Bit Probabilities**

The correlated Gaussian noise model described in this section uses soft decision information in the updating of the noise probability distribution function. The metric generator, upon receiving the coded bit probabilities from the previous iteration, calculates the average value of the coded bits from (3.19) as:

$$m_t^{(k)} = E \{ d_t^{(k)} \} = P_r \{ d_t^{(k)} = 1 \} - P_r \{ d_t^{(k)} = -1 \}, \quad k \in \{1, \ldots, K\}.$$  \hfill (3.36)

These average values of the coded bits are used to generate a soft estimate of the noise through an equation similar to (3.35) as:

$$\hat{n}_t^{(k)} = r_t^{(k)} - m_t^{(k)} - \rho \sum_{i=1 \atop i \neq k}^{K} m_t^{(i)}.$$  \hfill (3.37)

These soft noise estimates are used in place of $n_t^{(k)}$ in the calculation of the mean (3.32) for the input metric of the Turbo Code decoders. The iterative receiver proceeds to update the probability distribution of the MAI and the noise in the
rest of the iterations.

### 3.3.4 Extending The Correlated Gaussian Noise Model

The hard and soft decisions on the coded bits described in the previous section both use the conditional probability distribution $p\left( n_t^{(k)}|n_t^{(1)}, \ldots, n_t^{(K)} \right)$ in place of $p\left( n_t^{(k)} \right)$, where the conditional probability is calculated from the estimates of the noise $n_t^{(k)}$. However, the estimates of the noise are not exact but are random variables. As such, the noise estimate $\hat{n}_t^{(k)}$ can be approximated by a Gaussian distribution:

$$\hat{n}_t^{(k)} \sim N\left( \hat{n}_t^{(k)}, v_t^{2(k)} + \rho^2 \sum_{i=1, i \neq k}^{K} v_t^{2(i)} \right),$$

(3.38)

where $v_t^{2(k)}$ is the variance of the coded bits given in (3.20) and $\hat{n}_t^{(k)}$ is given by (3.37), which is independent on whether hard or soft estimates of the coded bits are utilized to create the noise estimates.

These distributions of the noise estimates of the $K - 1$ other users can be used to further update the probability distribution of the noise for the $k^{th}$ user. The updated noise probability distribution $p\left( \bar{n}_t^{(k)} \right)$ for user $k$ is obtained by integrating the product of the conditional probability distribution (3.29) and the joint probability distribution $p\left( \bar{n}_t^{(1)}, \ldots, \bar{n}_t^{(K)} \right)$ to yield:

$$p\left( n_t^{(k)} \right) = \int_{\bar{n}_t^{(1)}}^{\ldots} \int_{\bar{n}_t^{(K)}}^{\ldots} p\left( \bar{n}_t^{(1)}, \ldots, \bar{n}_t^{(K)} \right) \times$$

$$p\left( n_t^{(k)}|n_t^{(1)}, \ldots, n_t^{(K)} = \bar{n}_t^{(K)} \right) d\bar{n}_t^{(1)} \ldots d\bar{n}_t^{(K)}. \quad (3.39)$$
Since both the conditional and joint probabilities distributions are Gaussian, the probability distribution \( p(\hat{n}_t^{(k)}) \) will also be Gaussian. Thus, only the mean and variance need to be calculated for \( p(\hat{n}_t^{(k)}) \). The mean is calculated as:

\[
E \{ n_t^{(k)} \} = \int_{\hat{n}_t^{(1)}}^{\hat{n}_t^{(K)}} p(\hat{n}_t^{(1)}, \ldots, \hat{n}_t^{(K)}) \times E \{ n_t^{(k)} | n_t = \hat{n}_t^{(1)}, \ldots, n_t^{(K)} = \hat{n}_t^{(K)} \} d\hat{n}_t^{(1)} \ldots d\hat{n}_t^{(K)},
\]

where \( E \{ n_t^{(k)} | n_t = \hat{n}_t^{(1)}, \ldots, n_t^{(K)} = \hat{n}_t^{(K)} \} \) is the conditional mean \( \tilde{\mu}_t^{(k)} \) given in (3.30). Expanding \( E \{ n_t^{(k)} \} \) yields:

\[
\tilde{\mu}_t^{(k)} = E \{ n_t^{(k)} \} = \frac{\rho}{(K-2)\rho + 1} \int_{\hat{n}_t^{(1)}}^{\hat{n}_t^{(K)}} \int_{\hat{n}_t^{(1)}}^{\hat{n}_t^{(K)}} \sum_{i=1}^{K} \bar{n}_t^{(i)} p(\bar{n}_t^{(1)}, \ldots, \bar{n}_t^{(K)}) d\bar{n}_t^{(1)} \ldots d\bar{n}_t^{(K)}
\]

\[
= \frac{\rho}{(K-2)\rho + 1} \left[ \int_{\hat{n}_t^{(1)}}^{\hat{n}_t^{(K)}} \int_{\hat{n}_t^{(1)}}^{\hat{n}_t^{(K)}} \bar{n}_t^{(K)} p(\bar{n}_t^{(1)}, \ldots, \bar{n}_t^{(K)}) d\bar{n}_t^{(1)} \ldots d\bar{n}_t^{(K)} \right] \]

\[
= \frac{\rho}{(K-2)\rho + 1} \left[ E \{ \bar{n}_t^{(1)} \} + \ldots + E \{ \bar{n}_t^{(K)} \} \right]
\]

\[
= \frac{\rho}{(K-2)\rho + 1} \sum_{i=1}^{K} \hat{n}_t^{(i)}. \tag{3.41}
\]
The variance is calculated as:

\[
\bar{\sigma}_t^2 (k) = E \left\{ \left( n_t(k) - E \{ n_t(k) \} \right)^2 \right\} = E \{ n_t^2(k) \} - \left( E \{ n_t(k) \} \right)^2, \tag{3.42}
\]

where

\[
E \{ n_t^2(k) \} = \int_{n_t^{(1)}}^{\bar{n}_t^{(1)}} \int_{n_t^{(K)}}^{\bar{n}_t^{(K)}} p(\bar{n}_t^{(1)}, \ldots, \bar{n}_t^{(K)}) \times \nonumber
E \{ n_t^2(k) | n_t^{(1)} = \bar{n}_t^{(1)}, \ldots, n_t^{(K)} = \bar{n}_t^{(K)} \} \ d\bar{n}_t^{(1)} \ldots d\bar{n}_t^{(K)}. \tag{3.43}
\]

The conditional expectation in the above equation is determined by the equation:

\[
\bar{\sigma}^2 = E \left\{ n_t^2(k) | n_t^{(1)} = \bar{n}_t^{(1)}, \ldots, n_t^{(K)} = \bar{n}_t^{(K)} \right\} - \left[ \bar{\mu}_t^{(k)} \right]^2, \tag{3.44}
\]

where \( \bar{\mu}_t^{(k)} \) and \( \bar{\sigma}^2 \) are the conditional mean and variance in (3.30) and (3.31), respectively.

Equation (3.43) is expanded as:

\[
E \{ n_t^2(k) \} = \int_{n_t^{(1)}}^{\bar{n}_t^{(1)}} \int_{n_t^{(K)}}^{\bar{n}_t^{(K)}} p(\bar{n}_t^{(1)}, \ldots, \bar{n}_t^{(K)}) \left( \bar{\sigma}^2 + \left[ \bar{\mu}_t^{(k)} \right]^2 \right) d\bar{n}_t^{(1)} \ldots d\bar{n}_t^{(K)} \nonumber
= \sigma^2 \left[ \frac{(1 - \rho)(1 + (K - 1)\rho)}{(1 + (K - 2)\rho)} \right] + \left[ \frac{\rho}{(K - 2)\rho + 1} \right]^2 \times \nonumber
\int_{n_t^{(1)}}^{\bar{n}_t^{(1)}} \int_{n_t^{(K)}}^{\bar{n}_t^{(K)}} \left[ \sum_{i=1}^{K} \hat{n}_t^{(i)} \right]^2 p(\bar{n}_t^{(1)}, \ldots, \bar{n}_t^{(K)}) \ d\bar{n}_t^{(1)} \ldots d\bar{n}_t^{(K)} \nonumber
= \sigma^2 \left[ \frac{(1 - \rho)(1 + (K - 1)\rho)}{(1 + (K - 2)\rho)} \right] + \left[ \frac{\rho}{(K - 2)\rho + 1} \right]^2 \times \nonumber
\]
\begin{align*}
\int_{\hat{n}_t^{(1)}}^{\cdots} \int_{\hat{n}_t^{(K)}} \left[ \sum_{i=1}^{K} \hat{n}_t^{2(i)} + 2 \sum_{i=1}^{K-1} \sum_{j=2}^{K} \hat{n}_t^{(i)} \hat{n}_t^{(j)} \right] \times \\
p \left( \hat{n}_t^{(1)}, \cdots, \hat{n}_t^{(K)} \right) d\hat{n}_t^{(1)} \cdots d\hat{n}_t^{(K)}
\end{align*}

\begin{align*}
= \sigma^2 \left[ \frac{(1-\rho)(1+(K-1)\rho)}{(1+(K-2)\rho)} \right] + \left[ \frac{\rho}{(K-2)\rho+1} \right]^2 \times \\
\sum_{i=1}^{K} E \left\{ \hat{n}_t^{2(i)} \right\} + 2 \sum_{i=1}^{K-1} \sum_{j=2}^{K} E \left\{ \hat{n}_t^{(i)} \hat{n}_t^{(j)} \right\} .
\end{align*}

(3.45)

Substituting (3.41) and (3.45) into (3.42) and expanding yields the variance of the noise as:

\begin{align*}
\hat{\sigma}_t^{2(k)} &= \sigma^2 \left[ \frac{(1-\rho)(1+(K-1)\rho)}{(1+(K-2)\rho)} \right] \\
&+ \left[ \frac{\rho}{(K-2)\rho+1} \right]^2 \left[ \sum_{i=1}^{K} E \left\{ \hat{n}_t^{2(i)} \right\} + 2 \sum_{i=1}^{K-1} \sum_{j=2}^{K} E \left\{ \hat{n}_t^{(i)} \hat{n}_t^{(j)} \right\} \right] \\
&- \left[ \frac{\rho}{(K-2)\rho+1} \right]^2 \left[ \sum_{i=1}^{K} \hat{n}_t^{(i)} \right]^2 \\
&= \sigma^2 \left[ \frac{(1-\rho)(1+(K-1)\rho)}{(1+(K-2)\rho)} \right] + \left[ \frac{\rho}{(K-2)\rho+1} \right]^2 \times \\
&\left[ \sum_{i=1}^{K} \left( \hat{v}_t^{2(i)} + \rho^2 \sum_{j=1}^{K} \hat{v}_t^{2(j)} + \hat{\sigma}_t^{2(i)} \right) + 2 \sum_{i=1}^{K-1} \sum_{j=2}^{K} E \left\{ \hat{n}_t^{(i)} \hat{n}_t^{(j)} \right\} \\
&- \left( \sum_{i=1}^{K} \hat{\sigma}_t^{2(i)} + 2 \sum_{i=1}^{K-1} \sum_{j=2}^{K} \hat{n}_t^{(i)} \hat{n}_t^{(j)} \right) \right] .
\end{align*}
CHAPTER 3. ITERATIVE MULTI-USER CDMA RECEIVER

\[
\begin{align*}
&= \sigma^2 \left[ \frac{(1 - \rho)(1 + (K - 1)\rho)}{(1 + (K - 2)\rho)} \right] + \left[ \frac{\rho}{(K - 2)\rho + 1} \right]^2 \times \\
&\left[ \sum_{i=1 \atop i \neq k}^{K} v_t^2(i) + \rho^2 \sum_{j=1 \atop j \neq i}^{K} v_t^2(j) \right] + 2 \sum_{i=1 \atop i \neq k}^{K-1} \sum_{j=2 \atop j \neq k}^{K} E \left\{ \tilde{n}_t(i) \tilde{n}_t(j) \right\} \\
&- 2 \sum_{i=1 \atop i \neq k}^{K-1} \sum_{j=2 \atop j \neq k}^{K} \tilde{n}_t(i) \tilde{n}_t(j)
\end{align*}
\]

(3.46)

where \( E \left\{ \tilde{n}_t^2(i) \right\} = v_t^2(i) + \rho^2 \sum_{j=1 \atop j \neq i}^{K} v_t^2(j) + \tilde{n}_t^2(i) \) has been used from (3.38).

The term \( E \left\{ \tilde{n}_t(i) \tilde{n}_t(j) \right\} \) is calculated as:

\[
\begin{align*}
\tilde{n}_t(i) \tilde{n}_t(j) &= \left( r_t(i) - d_t(i) - \rho \sum_{k=1 \atop k \neq i}^{K} d_t(k) \right) \left( r_t(j) - d_t(j) - \rho \sum_{l=1 \atop l \neq j}^{K} d_t(l) \right) \\
&= r_t(i) r_t(j) - r_t(i) \left[ d_t(j) + \rho \sum_{l=1 \atop l \neq j}^{K} d_t(l) \right] - r_t(j) \left[ d_t(i) + \rho \sum_{k=1 \atop k \neq i}^{K} d_t(k) \right] \\
&+ d_t(i) d_t(j) + \rho d_t(i) \sum_{k=1 \atop k \neq i}^{K} d_t(k) + \rho d_t(j) \sum_{k=1 \atop k \neq i}^{K} d_t(k) + \rho^2 \sum_{k=1 \atop k \neq i}^{K} \sum_{l=1 \atop l \neq j}^{K} d_t(k) d_t(l) \\
&= r_t(i) r_t(j) - r_t(i) \left[ d_t(j) + \rho \sum_{l=1 \atop l \neq j}^{K} d_t(l) \right] - r_t(j) \left[ d_t(i) + \rho \sum_{k=1 \atop k \neq i}^{K} d_t(k) \right] \\
&+ d_t(i) d_t(j) + \rho d_t^2(i) + \rho d_t(i) \sum_{l=1 \atop l \neq j}^{K} d_t(l) + \rho d_t(j) \sum_{k=1 \atop k \neq i}^{K} d_t(k) + \rho^2 \sum_{k=1 \atop k \neq i, j}^{K} d_t(k) d_t(l)
\end{align*}
\]
\begin{align*}
&+ 2 \sum_{k=1 \atop k \neq i, j}^{K-1} \sum_{l=2 \atop l \neq i, j}^{K} d_t^{(k)} d_t^{(l)} \\
= & \quad r_t^{(i)} r_t^{(j)} - r_t^{(i)} \left[ d_t^{(j)} + \rho \sum_{l=1 \atop l \neq j}^{K} d_t^{(l)} \right] - r_t^{(j)} \left[ d_t^{(i)} + \rho \sum_{k=1 \atop k \neq i}^{K} d_t^{(k)} \right] \\
&+ \rho \left[ d_t^{2 (i)} + d_t^{2 (j)} + \rho \sum_{k=1 \atop k \neq i, j}^{K} d_t^{2 (k)} \right] \\
&+ \left( \rho^2 + \rho \right) \left[ d_t^{(j)} \sum_{k=1 \atop k \neq i, j}^{K} d_t^{(k)} + d_t^{(i)} \sum_{l=1 \atop l \neq j}^{K} d_t^{(l)} \right] \\
&+ \left( \rho^2 + 1 \right) d_t^{(i)} d_t^{(j)} + 2 \rho^2 \sum_{k=1 \atop k \neq i, j}^{K-1} \sum_{l=2 \atop l \neq i, j, l \geq k}^{K} d_t^{(k)} d_t^{(l)}.
\end{align*}

Taking the expectation of \( \tilde{n}_t^{(i)} \tilde{n}_t^{(j)} \) yields:

\begin{align*}
E \left\{ \tilde{n}_t^{(i)} \tilde{n}_t^{(j)} \right\} &= r_t^{(i)} r_t^{(j)} - r_t^{(i)} \left[ E \left\{ d_t^{(j)} \right\} + \rho \sum_{l=1 \atop l \neq j}^{K} E \left\{ d_t^{(l)} \right\} \right] \\
&\quad - r_t^{(j)} \left[ E \left\{ d_t^{(i)} \right\} + \rho \sum_{k=1 \atop k \neq i}^{K} E \left\{ d_t^{(k)} \right\} \right] \\
&+ \rho \left[ E \left\{ d_t^{2 (i)} \right\} + E \left\{ d_t^{2 (j)} \right\} + \rho \sum_{k=1 \atop k \neq i, j}^{K} E \left\{ d_t^{2 (k)} \right\} \right] \\
&+ \left( \rho^2 + \rho \right) \left[ \sum_{k=1 \atop k \neq i, j}^{K} E \left\{ d_t^{(j)} d_t^{(k)} \right\} + \sum_{l=1 \atop l \neq j}^{K} E \left\{ d_t^{(i)} d_t^{(l)} \right\} \right] \\
&+ \left( \rho^2 + 1 \right) E \left\{ d_t^{(i)} d_t^{(j)} \right\} + 2 \rho^2 \sum_{k=1 \atop k \neq i, j}^{K-1} \sum_{l=2 \atop l \neq i, j, l \geq k}^{K} E \left\{ d_t^{(k)} d_t^{(l)} \right\}.
\end{align*}
where the independence of the coded bit probabilities has been utilized.

The probability distribution of the metric \( p \left( r_t^{(k)} | d_t^{(k)} = d \right), d \in \{-1, 1\}, \) for the \( k^{th} \) user, with the generation of the updated noise probability distribution and the continued updating of the MAI, is a Gaussian random variable with the mean and
variance:

\[
\mu_t^{(k)} = d + \mu_{t, MAI}^{(k)} + \mu_t^{(k)} \\
= d + \rho \sum_{\substack{k=1 \\ k \neq i}}^{K} \left[ Pr \left\{ d_t^{(i)} = 1 \right\} - Pr \left\{ d_t^{(i)} = -1 \right\} \right] \\
+ \frac{\rho}{(K-2)\rho + 1} \sum_{\substack{k=1 \\ k \neq i}}^{K} n_t^{(i)} \\
\sigma_t^2(k) = \sigma_{t, MAI}^2(k) + \sigma_t^2(k),
\]

(3.48)

(3.49)

where the continuous Gaussian approximation decoder derived in 3.3.1 has been utilized to update probability distribution of the MAI. The terms \( \mu_{t, MAI}^{(k)} \) and \( \sigma_{t, MAI}^2(k) \) are the MAI Gaussian distribution’s mean and variance given in (3.21) and (3.22).
Chapter 4

Numerical Results

4.1 Simulation Setup

The bit error rate (BER) performance of the various iterative multi-user CDMA receivers derived in this thesis is presented in this chapter. The figures shown in this chapter are the average BER performance taken over all the users present in the system. The complexity of these various iterative receivers is also examined.

The results presented in this chapter are obtained by simulating the communication system described in chapter 2 over an AWGN channel. Each point on the figures is obtained by transmitting a minimum of one million bits per user to obtain a minimum of 100 errors per user.

Each user in the communication system employs a rate $R = 1/3$ Turbo Encoder, as shown in Figure 2.3, to provide the needed channel coding for the Turbo Code decoder. The generator polynomials $g_1 = 37_8$ and $g_2 = 21_8$ with $P = 4$ memory elements are used for both RSC encoders of the Turbo Encoder. The Turbo Code
decoder described in section 3.2 is used for the $K$ individual FEC decoders in Figure 3.1.

The simulations are performed using the block sizes of $M = 192$ bits and $M = 600$ bits$^1$. Most of the simulations are run with $K = 5$ and $K = 2$ users to illustrate the effect the number of users has on the performance of the iterative decoders.

The results for the decoders utilizing continuous approximation of the MAI and discrete analysis of the MAI are compared against the conventional decoder. Each user in the conventional decoder is decoded separately, performing Turbo decoding iterations only and treating the MAI as additional additive noise.

All of the iterative multi-user CDMA receivers described in the previous chapter are derived based on the simplification that the cross-correlation is constant between all users and is time invariant, given in equation (3.18). Figure 4.1 shows the BER performance of the continuous Gaussian approximation of the MAI decoder versus the channel SNR for various values of $\rho$ after the 5$^{th}$ iteration of the multi-user decoder, only 1 iteration of the Turbo decoder, for $K = 5$ users and a block size of $M = 192$. The iterations of the multi-user decoder are referred to as “MUit” in the figures. Iterations for the individual Turbo Code decoders are referred to as “TDit” in the figures.

The figure shows that as the amount of correlation increases, the performance of the continuous Gaussian approximation decoder becomes worse. This is expected since there is more interference that needs to be removed as $\rho$ increases. Most of

---

$^1$The block size of $M = 192$ bits corresponds to the IS-95 delay specification of 20 ms, assuming a data rate of 9.6 Kbits/sec. The larger block size of $M = 600$ bits is selected to show the effect that block size has on the performance of the decoders, while increasing the decoding delay slightly.
Figure 4.1: Continuous Approximation of MAI with $K = 5$ Users and Block Size $M = 192$ for Various Values of $\rho$

the simulations are performed using spreading signals of $N = 7$ chips for a cross-correlation value of $\rho = 1/7$. The parameters of $K = 5$ users and chip length $N = 7$ represent a typical loaded CDMA system, as used in [13].

The BER performance results presented in this chapter, show that the MAI interference is essentially removed after 5 iterations which is a typical number of iterations for Turbo Codes. The continuous Gaussian approximation of the MAI decoder and discrete analysis decoder give similar results with the continuous Gaussian approximation decoder being less complex. The correlated Gaussian noise model decoders perform similarly to the continuous Gaussian approximation decoder while the extended correlated Gaussian noise model performs slightly better. However, the correlated and extended correlated Gaussian noise models are more complex than the continuous Gaussian approximation decoder.
4.2 Continuous Gaussian Approximation of MAI BER Performance

Figures 4.2 and 4.3 show the average BER performance of the iterative multi-user CDMA decoder for $K = 5$ users utilizing the probability metric based on approximating the MAI as a Gaussian random variable as derived in section 3.3.1. Figure 4.2 illustrates the BER performance for a block size of $M = 192$ bits while Figure 4.3 shows the BER performance for a block size of $M = 600$ bits.

Both figures illustrate the performance of the iterative decoder at the 5th and 10th iteration of the multi-user decoder. Only one iteration of the Turbo Code decoder is performed for these simulations. The BER performance results are compared to the performance of the conventional decoder mentioned previously and the performance of the system when only a single-user is present in the system using Turbo Codes for FEC. The single-user case is equivalent to the spreading signals of all the users in the CDMA system being orthogonal, i.e., $\rho = 0$.

Figures 4.2 and 4.3 show that after the 10th iteration of the iterative multi-user decoder using the continuous Gaussian approximation probability distribution near single-user performance is achieved at both block sizes. The figures also show that there is still considerable improvement by the iterative decoder from the 5th to the 10th iteration, about 0.15dB at a BER of $10^{-3}$ for both block sizes of $M = 192$ and $M = 600$ bits. However, there is little improvement at higher signal-to-noise ratio (SNR) values for a block size of $M = 600$. There is also about 0.3dB improvement of the iterative multi-user decoder over the conventional decoder performing the
CHAPTER 4. NUMERICAL RESULTS

Figure 4.2: Continuous Approximation of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.3: Continuous Approximation of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$
same number of iterations.

Figures 4.4 and 4.5 show the BER performance of the iterative multi-user decoder for $K = 2$. In this case, only one user is included in the MAI term. Near single-user performance is achieved by the iterative decoder at the 10\textsuperscript{th} iteration although not as tight as for the $K = 5$ user results. This slightly worse performance is due to the approximation that the MAI is Gaussian, with the approximation being more accurate as the number of users increases. The conventional decoder performs better for $K = 2$ users than for $K = 5$ users, being worse than the iterative decoder by less than 0.1dB while performing better than the iterative decoder. This was expected as the $K = 2$ user case is closer to the ideal single-user system than the $K = 5$ user case.

### 4.3 Discrete Analysis of MAI BER Performance

The average BER performance when the iterative multi-user decoder utilizes the discrete analysis of the MAI probability metric, as derived in section 3.3.2, is shown in Figures 4.6 and 4.7 for block sizes of $M = 192$ and $M = 600$ bits, respectively. These figures are similar to the ones for the continuous Gaussian approximation decoder. Near single-user performance has been achieved by the 10\textsuperscript{th} iteration for both block sizes. At a BER of $10^{-3}$, the 10\textsuperscript{th} iteration is about 0.15dB better than the 5\textsuperscript{th} iteration and about 0.3dB better than the conventional decoder with 10 Turbo Code iterations for both block sizes. However, there is little improvement from the 5\textsuperscript{th} to the 10\textsuperscript{th} iteration at larger SNR values for a block size of $M = 600$.

Figures 4.8 and 4.9 show the average BER performance of the discrete analysis
Figure 4.4: Continuous Approximation of MAI with $K = 2$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.5: Continuous Approximation of MAI with $K = 2$ Users, Block Size $M = 600$ and Chip Length $N = 7$
CHAPTER 4. NUMERICAL RESULTS

Figure 4.6: Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.7: Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$
of the MAI decoder for $K = 2$ users at a block size of $M = 192$ and $M = 600$ bits. The figures show that near single-user performance is achieved by the $10^{th}$ iteration, although not as good as for the $K = 5$ user results due to the less accurate approximation of the probability distribution of the MAI term. As in the continuous approximation of the MAI, the conventional decoder performs better for $K = 2$ users than for $K = 5$ users.

4.4 Continuous Approximation Versus Discrete Analysis of MAI BER Performance

Figures 4.10 and 4.11 compare the results of the continuous Gaussian approximation of the MAI metric with the results of the discrete analysis of the MAI metric for the block sizes $M = 192$ and $M = 600$ bits. The $5^{th}$ iteration of the two iterative multi-user decoders is shown along with the performance of the single-user system after the $5^{th}$ iteration. These figures show that near single-user performance is achieved with only five iterations of the iterative decoders. They also show that the performance of the two decoders is similar when five users are transmitting over the AWGN channel.

The comparison of the results at the $5^{th}$ iteration of the continuous Gaussian approximation of the MAI metric and the discrete analysis of the MAI metric when only two users are using the communication system are shown in Figures 4.12 and 4.13 for the block sizes $M = 192$ and $M = 600$ bits. These figures show that the two iterative decoders perform similarly for $K = 2$ users at both block sizes. This
Figure 4.8: Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.9: Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 600$ and Chip Length $N = 7$
Figure 4.10: Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.11: Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$
is expected since the two metric derivations are similar when two users are present in the system, having only one user making up the MAI interference term.

Reed et al. [13] derived an iterative multi-user decoder using Turbo Codes for a CDMA system with a set of assumptions on the system similar to the assumptions for the system used in this thesis. Their decoder generated a conditional probability distribution function for the individual Turbo Code decoder of user $k$ by evaluating the joint probability distribution $p(r_t|d_t)$ over all possible coded sequences $d_t$ in which $d_t^{(k)}$ was fixed at a certain value. This decoder has exponential complexity with the number of users transmitting in the system, like the optimal detector [8]. The simulations performed in the Reed et al. [13] paper are done for a synchronous system with the parameters: chip length $N = 7$, block size $M = 200$, and $K = 5$ users while utilizing random spreading signals. In [13], simulation results are presented for three multi-user system iterations, each with four Turbo Code iterations and the corresponding average BER performance results are compared with a single-user system using Turbo Code for FEC, with four iterations. At an average BER of $10^{-3}$, the Reed et al. iterative multi-user decoder, after three system iterations, is about 0.25dB worse than the single-user performance.

For the sake of comparison, Figure 4.14 presents the results of the continuous Gaussian approximation and discrete analysis of the MAI decoders that are derived in this thesis with similar parameters as the Reed et al. work, the only difference being a slightly smaller block size $M = 192$ being used. The simulation results presented show the performance of the decoders after the third multi-user iteration, each with only one Turbo Code iteration, compared with the single-user system
CHAPTER 4. NUMERICAL RESULTS

Figure 4.12: Continuous Approximation Versus Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.13: Continuous Approximation Versus Discrete Analysis of MAI with $K = 2$ Users, Block Size $M = 600$ and Chip Length $N = 7$
Figure 4.14: Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$ after four Turbo Code iterations. This figure shows that the two decoders resulted in similar performances that are about 0.2dB worse than the single-user system at an average BER of $10^{-3}$. Thus, the less complex decoders presented in this thesis perform similarly and slightly better, at fewer iterations, than the more complex solution of Reed et al. [13]. This result occurs from the way these sub-optimal receivers perform the decoding operation. The receiver introduced by Reed et al. [13] performs the updating operation of their iterative decoder over a sequence of received bits resulting in the exponential complexity. However, the proposed receivers in this thesis perform the updating operation on a per bit basis that results in a much lower complexity.
4.5 Correlated Gaussian Noise Model BER Performance

The correlated Gaussian noise model derived in section 3.3.3 described two ways of updating the probability distribution function of the correlated Gaussian noise. These ways are based on making a hard or soft decision on the coded bit probabilities determined from the previous iteration. Since the probability distribution function of the noise is based on the coded bit probabilities, a number of iterations are performed in which the probability distribution of the MAI is updated. The probability distribution of the MAI was chosen as the continuous Gaussian approximation in order to keep the complexity low.

Figures 4.15 and 4.16 compare the average BER performance of the iterative decoder utilizing the continuous Gaussian approximation for the MAI and hard and soft decisions on the coded bit probabilities for the Gaussian noise for the block sizes of $M = 192$ and $M = 600$, respectively. The iterations of the decoder that include the updating of the probability distribution of the MAI only are referred to as “MAIit” in the figures while “MUit” refers to the iterations that also include the updating of the noise distribution. The figures also show the BER performance results for both the continuous Gaussian approximation decoder and the single-user system after the 6th iteration.

The figures show that the updating of the noise probability distribution during the 6th iteration performs similarly to the continuous Gaussian approximation decoder at both block sizes, with the soft decision decoder performing slightly better
at higher SNR values. Near single-user performance is achieved at both block sizes with the decoders achieving a better performance at the block size of $M = 600$.

### 4.6 Extended Correlated Gaussian Noise Model

**BER Performance**

The average BER performance of the extended correlated Gaussian noise model derived in section 3.3.4 for a block size of $M = 192$ and $M = 600$ is shown in Figures 4.17 and 4.18. Figure 4.17 shows that the extended correlated Gaussian noise decoder performs better than the continuous Gaussian approximation decoder for the 6\textsuperscript{th} iteration at a block size of $M = 192$. The extended correlated Gaussian noise decoder at the higher SNR values achieves single-user performance. The BER performance of the two decoders is almost identical for a block size of $M = 600$, as shown in Figure 4.18, since single-user performance has been achieved at the 6\textsuperscript{th} iteration by both decoders.

The BER performance of the extended correlated Gaussian noise decoder for a highly loaded system with $K = 7$ users at a block size of $M = 192$ is shown in Figure 4.19. This figure shows that the continuous Gaussian approximation decoder performs much better at higher SNR values than the extended correlated Gaussian noise decoder for a highly loaded CDMA system. The continuous Gaussian approximation of the MAI becomes more valid when the number of users increases. However, the assumptions used for the extended correlated Gaussian noise model are not as valid when the number of users increases resulting in a worse performance.
Figure 4.15: Hard Versus Soft Decision of Noise with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.16: Hard Versus Soft Decision of Noise with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$
than the continuous Gaussian approximation decoder.

4.7 Complexity Analysis

The computational complexity of the various iterative multi-user CDMA decoders is examined in Appendix A. The result of this analysis is discussed in this section.

The computational complexity of the iterative decoders is linearly dependent on the number of iterations performed for the decoder. The different iterations are multi-user iterations, referred to as “MUit”, Turbo Code decoder iterations, referred to as “TDit”, and “MAIit” refers to the MAI iterations in the correlated Gaussian noise models. These different iterations are the same that are used previously in this chapter.

The computational complexity of the iterative decoders is also dependent on the number of users $K$ in the system. From (A.14), the continuous Gaussian approximation of the MAI decoder has polynomial complexity proportional to $K^2$ per multi-user iteration. The complexity of the discrete analysis of the MAI decoder in (A.15) shows that it has polynomial complexity proportional to $K^3$ per multi-user iteration. Equation (A.17) shows that the correlated Gaussian noise model decoder has polynomial complexity that is proportional to $K^2$ per multi-user and MAI iteration. The computational complexity of the extended correlated Gaussian noise model decoder, from (A.20), has the same complexity as the continuous Gaussian approximation decoder for the MAI iterations and has polynomial complexity proportional to $K^4$ per multi-user iteration thereafter.

Figures 4.10 and 4.11 show the comparison between the continuous Gaussian
Figure 4.17: Extended Correlated Gaussian Noise Model with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

Figure 4.18: Extended Correlated Gaussian Noise Model with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$
Figure 4.19: Extended Correlated Gaussian Noise Model with $K = 7$ Users, Block Size $M = 192$ and Chip Length $N = 7$

approximation and the discrete analysis of the MAI decoder when $K = 5$ users are utilizing the system for block sizes of $M = 192$ and $M = 600$. These results are performed with only 1 Turbo Code decoder iteration and 5 multi-user decoder iterations. The block size does not affect the relative computational complexity per input bit. The figures show that both decoders have a similar BER performance that is near to the single-user case. Table 4.1 compares the computational complexity per user of these two decoders per input bit relative to the single-user Turbo Code system. The single-user system performed TDit = 5 Turbo decoding iterations. Equations (A.13), (A.14), (A.15), and (A.16) are used to compute the relative complexity. Table 4.1 shows that the discrete analysis decoder is 30% more complex than the continuous Gaussian approximation decoder.
CHAPTER 4. NUMERICAL RESULTS

Table 4.1: Relative Computational Complexity Per User For Continuous Gaussian Approximation and Discrete Analysis of MAI Receivers

<table>
<thead>
<tr>
<th></th>
<th>$K = 2$ Users</th>
<th>$K = 5$ Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-User System</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Continuous Gaussian Approximation of MAI</td>
<td>1.45</td>
<td>1.51</td>
</tr>
<tr>
<td>Discrete Analysis of MAI</td>
<td>1.47</td>
<td>1.81</td>
</tr>
</tbody>
</table>

Table 4.1 also gives the relative computational complexity per user when only $K = 2$ users are present in the system. The performance results for these parameters are shown in Figures 4.12 and 4.13 for the block sizes of $M = 192$ and $M = 600$. The discrete analysis decoder is only 2% more complex than the continuous Gaussian approximation decoder. The complexity results for $K = 5$ and $K = 2$ users show that the discrete analysis decoder is increasingly more complex than the continuous Gaussian approximation decoder when the number of users increases. This complexity difference is expected since the discrete analysis decoder has to perform the computationally intensive convolution operation in order to compute the probability distribution $p\left( r_t^{(k)} | d_t^{(k)} \right)$. Thus, the continuous Gaussian approximation decoder is the better decoder since it performs similarly to the discrete analysis decoder at a smaller computational complexity.

The complexity results also show that the continuous Gaussian approximation decoder requires about 50% more calculations than the single-user case. This is a minimal amount of extra computations for the CDMA multi-user iterative decoder over the single-user system as compared to the exponential complexity for the optimal coded multi-user decoder of Giallorenzi and Wilson [31] and the sub-optimal coded multi-user decoder of Reed et al. [13].
Table 4.2 gives the relative computational complexity per user for $K = 5$ users of the correlated and extended Gaussian noise models as compared to the single-user Turbo Code system. The decoding of the correlated and extended Gaussian noise models performed MAIit = 5, MUit = 1, and TDit = 1 iterations while the single-user performed TDit = 6 Turbo decoding iterations. The relative computational complexity per user of the continuous Gaussian approximation decoder is also given for MUit = 6 and TDit = 1. Equations (A.13), (A.14), (A.17), and (A.20) are used to compute these values. The performance results for these decoders with $K = 5$ users are shown in Figures 4.15 and 4.16 for the correlated Gaussian noise model at block sizes $M = 192$ and $M = 600$. Figures 4.17 and 4.18 show the performance results of the extended correlated Gaussian noise model for $K = 5$ users at block sizes of $M = 192$ and $M = 600$.

The computational complexity results in Table 4.2 show that the continuous Gaussian approximation decoder and the correlated Gaussian noise model, for both hard and soft decisions, have similar complexity with $K = 5$ users while the soft decision correlated Gaussian noise model decoder performs slightly better, at a
block size of $M = 192$, than the other decoders. The extended correlated Gaussian noise model decoder performs slightly better than the continuous Gaussian approximation decoder for $K = 5$ and $M = 192$ while requiring only about 10% more computations.
Chapter 5

Conclusions

This thesis studies a number of different sub-optimal iterative multi-user receivers for CDMA, utilizing Turbo Codes for forward error correction purposes. The proposed decoders, differing by the probability distribution generated for use in each individual Turbo decoder, are presented. The BER performance of these iterative decoders are simulated and shown. The computational complexity of the decoders is also analyzed.

The continuous Gaussian approximation of the MAI decoder performed similarly to the discrete analysis of the MAI decoder for different number of users and block size. However, the continuous Gaussian approximation decoder is less complex than the discrete analysis decoder, requiring fewer computations. Thus, the continuous Gaussian approximation decoder is better than the discrete analysis decoder, yielding similar performance at a lower complexity. The continuous Gaussian approximation of the MAI decoder also performs similarly and slightly better than the more complex iterative decoder presented by Reed et al. [13].
The correlated Gaussian noise model decoder for both soft and hard decisions on the coded bit probabilities perform similarly to the equivalent continuous Gaussian approximation decoder, achieving near single user BER performance. However, the results show that the added computational complexity of the correlated noise model decoder is not justified compared to the continuous Gaussian approximation decoder.

The extended correlated Gaussian noise model decoder performs better than the correlated Gaussian noise model decoder and the continuous Gaussian approximation of the MAI decoder for small block sizes and for systems not loaded too heavily. The computational complexity of the extended correlated Gaussian noise model decoder is greater than the complexity of the continuous Gaussian approximation decoder.

5.1 Future Research Directions

The following future research directions in the study of the proposed iterative multi-user CDMA receivers with Turbo Codes for FEC are proposed:

- Extend the iterative multi-user decoders to include asynchronous bit and chip time intervals.

- Examine the effect of time variant cross-correlation values on the iterative multi-user decoders.

- Extend the iterative multi-user decoders to a RAKE receiver.

- Examine the BER performance for a fading channel model.
• Investigate the effect non-perfect power control has on BER performance of decoders.

• Further develop correlated noise models in order that updating of noise can take place from the first iteration.
Consider the iterative multi-user CDMA decoder shown in Figure 3.1. The complexity analysis of this decoder can be broken down into three parts. These parts include the complexity of the $K$ individual Turbo Code decoders, the complexity of the final hard decision, and the complexity of the metric generator. The complexity of the metric generator is dependent on the particular decoder being considered.

The number of numerical calculations derived in the following sections do not include the mathematical functions of “exp” and “ln”. It is assumed that these functions are implemented by using a look-up table, which does not constitute a calculation. The interleaving process is also not included in the number of calculations as a look-up table also accomplishes it.
APPENDIX A. COMPLEXITY CALCULATIONS

A.1 Turbo Code Complexity

The Turbo Code decoder is discussed in section 3.2. It takes the metric probabilities, 
\( p(r_i^{(k)}|d_i^{(k)}) \), generated by the metric generator and performs the modified BJCR algorithm before a hard decision is made on the transmitted bits. This section will review the BJCR algorithm to find its numerical complexity. The RSC components of the Turbo Code decoder have \( P \) memory elements.

The forward recursion values \( \alpha_r(m) \) of the BJCR algorithm are calculated, given in (3.2), as:

\[
\alpha_r(m) = \frac{\sum_{m'}\sum_{i=0}^1 \gamma_i(R_r, m', m)\alpha_{r-1}(m')}{\sum_m\sum_{m'}\sum_{i=0}^1 \gamma_i(R_r, m', m)\alpha_{r-1}(m')}.
\] (A.1)

This expression requires 3 multiplications to form each term in the numerator since the \( \gamma_i(R_r, m', m) \) values, from (3.5), consist of three terms when a branch in the trellis exists, requiring 2 multiplications. To form the first summation, 2 multiplications are needed. These calculations must be performed for each of the \( 2^P \) states of the trellis. A total of \( 3(2)(2^P) \) multiplications are required to calculate (A.1). Also, the first summation requires 1 addition for each of the \( 2^P \) states and the second summation requires \( 2^P - 1 \) additions. The calculation of the common denominator term can be omitted to reduce the computational complexity, resulting only in a negligible reduction in BER performance. Therefore, the total number of calculations required for \( \alpha_r(m) \) is \( 3(2^{P+1}) + 2^P + 2^P - 1 = 4(2^{P+1}) - 1 \).

From (3.3), the backward recursion values \( \beta_r(m) \) are given by the expression:

\[
\beta_r(m) = \frac{\sum_{m'}\sum_{i=0}^1 \gamma_i(R_{r+1}, m', m)\beta_{r+1}(m')}{\sum_m\sum_{m'}\sum_{i=0}^1 \gamma_i(R_{r+1}, m', m)\alpha_r(m')}.
\] (A.2)
Using a similar analysis as for the $\alpha_r(m)$ calculations, the $\beta_r(m)$ values also require $4(2^{P+1}) - 1$ calculations if the common denominator is omitted.

The forward and backward recursion values are used to generate the LLR values of the BJCR algorithm as:

$$\Lambda(b_r) = \ln \left( \frac{\sum_m \sum_{m'} \gamma_1(R_r, m', m) \alpha_{r-1}(m') \beta_r(m)}{\sum_m \sum_{m'} \gamma_0(R_r, m', m) \alpha_{r-1}(m') \beta_r(m)} \right). \quad (A.3)$$

The central term in (A.3) requires 4 multiplications and needs to be calculated for each of the $2(2^P)$ branches in the trellis. Each summation is performed over all the branches of the trellis with the same input bit, thus $2^P - 1$ additions are required. These additions are performed for both the numerator and denominator. One division is required to complete the expression. Therefore, $4(2^P)(2) + 2(2^P - 1) + 1 = 5(2^{P+1}) - 1$ calculations are required for (A.3).

The remaining calculations of the modified BJCR algorithm deal with the manipulation of the LLR values. First, the LLR values are converted to APP values, as in (3.13), requiring 1 addition and 1 division for each bit $\in \{0, 1\}$. These values are calculated once at the start of the RSC decoders and stored. Second, the calculation of the intrinsic values $I_r$ from (3.11) requires 1 division. Lastly, the extrinsic component of the LLR for use in the next RSC decoder, calculated from (3.11), requires 2 subtractions.

Therefore, each RSC decoder of the Turbo Code decoder requires $4(2^{P+1}) - 1 + 4(2^{P+1}) - 1 + 5(2^{P+1}) - 1 + 2(1+1) + 1 + 2 = 13(2^{P+1}) + 4$ calculations. The two-code Turbo Code decoder shown in Figure 3.2 requires $2(13(2^{P+1}) + 2) = 26(2^{P+1}) + 8$ calculations per input bit for each iteration of the Turbo Code decoder.
The iterative multi-user decoder requires the Turbo Code decoders to also supply LLR values for the parity bits. An additional \(5(2^{P+1}) - 1\) calculations are needed for each of the LLR parity bit values. The LLR values also need to be converted to coded bit probabilities. This conversion is done using (3.13), requiring 1 addition and 1 division for bit probability and each LLR. Thus, each Turbo Code decoder in the multi-user decoder requires \(26(2^{P+1})+8+2(5(2^{P+1})-1)+3(2)(2) = 36(2^{P+1})+18\) calculations for each input bit of each user.

### A.2 Final Hard Decision Complexity

The final hard decision is independent of the number of Turbo Code iterations and multi-user iterations as it is performed only during the last iteration. The hard decision can made with 1 calculation to decide if the LLR is positive or negative. Thus, the final hard decision only adds 1 calculation per bit for each user to the overall complexity.

### A.3 Metric Generator Complexity

The complexity of the metric generator is dependent on the particular metric being calculated. Sections 3.3.1 to 3.3.4 discuss a number of different probability metrics that the metric generator computes. The complexity of these metrics is discussed in the following sections. The next section gives the overall complexity of each decoder.
Continuous Gaussian Approximation of MAI

The metric generator calculates the probability metrics $p\left( r_t^{(k)} | d_t^{(k)} \right)$ for use in the Turbo Code decoders. For the continuous Gaussian approximation of the MAI decoder, the metric generator calculates the mean and variance of the systematic and parity bits for each input bit before calculating the probability metrics.

From (3.23), the calculation of the mean $\mu_t^{(k)}$ requires 1 addition plus the number of calculations required to generate the mean $\mu_{t, \text{MAI}}^{(k)}$ of the MAI. The mean of the MAI is calculated from (3.21) as:

$$
\mu_{t, \text{MAI}}^{(k)} = \rho \sum_{i=1, i\neq k}^{K} \left[ P_r \left\{ d_t^{(i)} = 1 \right\} - P_r \left\{ d_t^{(i)} = -1 \right\} \right].
$$

(A.4)

This expression requires 1 subtraction for each of the $K - 1$ users in the summation and $K - 2$ additions to add these terms together. The multiplication of $\rho$ requires 1 multiplication. Therefore, $1 + (K - 1) + (K - 2) + 1 = 2K - 1$ calculations are required to calculate the mean for each of the systematic and parity bits of each user.

The variance of the probability metrics, from (3.24), requires 1 addition plus the number of calculations required to generate the MAI variance:

$$
\sigma_{t, \text{MAI}}^{2(k)} = \rho^2 \sum_{i=1, i\neq k}^{K} \left( 1 - \left[ P_r \left\{ d_t^{(i)} = 1 \right\} - P_r \left\{ d_t^{(i)} = -1 \right\} \right]^2 \right).
$$

(A.5)

The MAI variance requires 2 subtractions and 1 multiplication for each of the $K - 1$ users that make up the MAI term. The summation requires $K - 2$ additions. The
term $\rho^2$ is assumed to be pre-calculated and stored so only 1 multiplication is required. A total of $1 + 3(K - 1) + (K - 2) + 1 = 4K - 3$ calculations are required for the variance of each systematic and parity bit of each user.

The metric generator calculates and stores the mean and variance of the systematic and parity bits for each user to be used in the generation of the probability metrics corresponding to one input bit. This process requires a total of $3(2K - 1) + 3(4K - 3) = 18K - 12$ calculations per multi-user iteration for each input bit of each user.

The metric generator also calculates the probability metrics that are used by the $K$ individual Turbo Code decoders in the calculation of $\gamma_i(R_r, m', m)$. The probability metrics, when the continuous Gaussian approximation of the MAI is used, are Gaussian random variables that are generated through the expression:

\[
p(r_{t}^{(k)}|d_{t}^{(k)} = d) = \frac{1}{\sqrt{2\pi\sigma_{t}^{2}(k)}} \exp \left[-\frac{(r_{t}^{(k)} - d - \mu_{t}^{(k)})^2}{2\sigma_{t}^{2}(k)} \right], \quad (A.6)
\]

where $d \in \{-1, 1\}$. The term $1/\sqrt{2\pi\sigma_{t}^{2}(k)}$ can be omitted from calculations since it has little effect on the BJCR algorithm. The numerator of the exponential term requires 2 subtractions and 1 multiplication while the denominator requires 1 multiplication. One division is also needed to compute the exponential term. Thus, 5 calculations are needed to generate each metric.

For each input bit that is transmitted, three metrics are required, one for each of the systematic and parity bits. These calculations also need to be done for both symbols $d \in \{-1, 1\}$. Thus, $3(2)(5) = 30$ total calculations are needed per
multi-user iteration for each input bit of each user.

**Discrete Analysis of MAI Complexity**

The metric generator, for the discrete analysis of the MAI decoder, needs to calculate the metric, given in (3.26), as:

\[
p(r_t^{(k)}|d_t^{(k)} = d) = \sum_m \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(r_t^{(k)} - d - pm)^2}{2\sigma^2} \right] \times P_{t,m}^{(k)}, \quad (A.7)
\]

where \( m \in \{- (K - 1), -(K - 3), \ldots, (K - 3), (K - 1)\} \) and \( d \in \{-1, 1\} \).

The terms of the length \( K \) sequence \( \{P_{t,m}^{(k)}\} \) are pre-calculated and stored before the metric values are computed. This sequence results from the convolution expressed in (3.25). The discrete convolution in (3.25) requires, \( K \geq 3 \),

\[
2 \sum_{m=1}^{\left\lfloor \frac{K-1}{2} \right\rfloor} \binom{K-1}{K-m} (K-2) + c \left( \frac{K-1}{2} \right) (K-2) \quad (A.8)
\]

multiplications\(^1\), where

\[
c = \begin{cases} 
1 : & K \text{ odd} \\
2 : & K \text{ even}
\end{cases}
\]

and

\[
\binom{a}{b} = \frac{a!}{b!(a-b)!}.
\]

\(^1\)The operators \( \lfloor \cdot \rfloor \) and \( \lceil \cdot \rceil \) represent the floor and ceiling operations, respectively.
The discrete convolution also requires, \( K \geq 3, \)

\[
2 \sum_{m=1}^{\lfloor \frac{K-1}{2} \rfloor} \left( \frac{K-1}{K-m} - 1 \right) + c \left( \frac{K-1}{K - \lceil \frac{K}{2} \rceil} - 1 \right)
\]  

(A.9)

additions. Thus, the total number of calculations required by the sequence \( \{P_{t,m}^{(k)}\} \)
is the addition of (A.8) and (A.9). The calculations for the discrete convolution are
not needed when \( K = 2 \) since only one interference user is present.

The calculation of the metric in (A.7) requires 2 subtractions and 2 multipli-
cations for each exponential term and 1 multiplication of \( P_{t,m}^{(k)} \) for each of the \( K \)terms in the summation. The constant \( 1/\sqrt{2\pi\sigma^2} \) can be omitted without affect-
ing the BJCR algorithm. The summation requires \( K - 1 \) additions to combine the \( K \) terms. Finally, 1 multiplication is needed to include the pre-calculated and stored common factor \( e^{2\sigma^2} \). These calculations need to be done for both bit values \( d \in \{-1, 1\} \) and for the systematic and parity bits of the corresponding input bit. Therefore, the computation of the metrics require \( 3(2)(5K + (K - 1) + 1) = 36K \)total calculations for each input bit of each user.

**Correlated Gaussian Noise Model Complexity**

For the iterative decoder utilizing the correlated Gaussian noise model, the metric generator needs to compute the mean and variance of the systematic and parity bits. This calculation is similar to the case of continuous Gaussian approximation for the MAI.

The metric generator must compute the mean and variance for this iterative decoder according to the equations given in (3.32) and (3.33). These values consist
of the mean and variance computed for the continuous Gaussian approximation decoder used to update the probability distribution of the MAI. The mean of the MAI was found to require $2K - 2$ calculations while the variance required $4K - 4$ calculations for each user and coded bit.

The mean of the updated noise probability is computed from (3.30) as:

$$\tilde{\mu}_t^{(k)} = \frac{\rho}{(K - 2)\rho + 1} \sum_{i \neq k} n_t^{(i)}.$$

The constant is pre-calculated and stored in order to reduce the decoder complexity. The summation requires $K - 2$ additions while 1 multiplication is needed for the constant term. The calculation for the mean of the metric (3.32) also requires 2 additions. Therefore, the computation of each metric’s mean requires a total of $(2K - 2) + K - 2 + 1 + 2 = 3K - 1$ calculations for each user.

The variance of the updated noise probability is a constant and can be pre-calculated and stored so it does not add to the complexity of the decoder. The computation of the variance of the metric requires 1 addition in order to add the two variances together. Thus, the variance of the metric requires $(4K - 4) + 1 = 4K - 3$ calculations for each user.

The calculation of the mean and variance for the probability metric $p \left( r_t^{(k)} | d_t^{(k)} \right)$ must be performed for the systematic and parity bits corresponding to the input bit of each user. The metric generator must perform a total of $3(3K - 1) + 3(4K - 3) = 21K - 12$ calculations for each user to compute the mean and variances for one input bit.

The noise values $n_t^{(i)}$ used in the generation of the mean for the updated noise
probability distribution are computed by the metric generator for all $K$ users and stored before the mean is calculated. The complexity of computation of the noise values is dependent on whether a hard or soft decision is made on the coded bit probabilities. For the noise values computed by making a hard decision, 1 calculation for each of the $K$ users is required to make the hard decision on the coded bit probabilities obtained from the previous iteration. The computation of the noise value using these hard decisions, given in (3.35), requires $K - 2$ additions and 1 multiplication for the MAI term. Two subtractions are required to finish the calculation of the noise value. The noise value must be computed for the systematic and parity bits for all $K$ users. Therefore, a total of $3(1 + K - 2 + 1 + 2)K = 3K^2 + 6K$ calculations are required to compute the noise values of all users per input bit.

When soft decisions are made on the coded bit probabilities, the average value of the coded bit probabilities is used in place of the bit value in the computation of the noise estimates. The calculation of the average value, from (3.36), requires 1 subtraction for each user. The metric generator calculates the average value for all $K$ users and stores them for use in the computation of the noise estimates. The noise value is computed in the same manner as above when hard decisions are used. Thus, since making a soft decision utilizes the same number of calculations as a hard decision, $3K^2 + 6K$ calculations are also needed to compute all the noise estimates of the $K$ users per input bit.

After the mean and variances are calculated, the metric generator must also compute the metric values for the systematic and parity bits of each user corresponding to one input bit. The metrics are computed by the equation in (A.6).
From the previous analysis of (A.6), a total of 30 calculations are required to compute the three probability metrics of each user for each input bit.

**Extended Correlated Gaussian Noise Model Complexity**

The calculation of the mean for the extended correlated Gaussian noise model is the same as that for the correlated Gaussian noise model. The calculation of the noise estimates \( \hat{n}_t^{(k)} \) and coded bit means \( m_t^{(k)} \) require \( 3K^2 + 6K \) calculations for all \( K \) users and for the three metrics of a single input bit. The computation of the mean \( \mu_t^{(k)} \) from (3.48) requires \( 9K^2 - 3K \) total calculations for all users per input bit, as found for the correlated Gaussian noise model.

The extended correlated Gaussian noise model computes a much different variance than the correlated Gaussian noise model. The variance is calculated, from (3.49), as:

\[
\sigma_t^2(k) = \sigma_{t,MAI}^2 + \bar{\sigma}_t^2(k),
\]

where the continuous Gaussian approximation of the MAI has been used. As previously derived, each MAI variance requires \( 4K - 4 \) calculations per user.

The variance of the correlated Gaussian noise is calculated from (3.46). The first term in (3.46) is a constant and is pre-calculated and stored. The constants \([\rho/((K - 2)\rho + 1)]^2\) and \(\rho^2\) and also pre-calculated and stored. The noise estimates \(\hat{n}_t^{(k)}\) have already been computed and stored in the calculation of the mean. The terms \(v_t^2(k)\) and \(E \{ \bar{n}_t^{(i)} \bar{n}_t^{(j)} \} \) are pre-calculated and stored for all \( K \) users by the metric generator prior to computing the noise variance.

The calculation of \(\rho^2 \sum_{j=1}^{K} v_t^2(j)\) of the second term in (3.46) requires \( K - 2 \) addi-
tions and 1 multiplication for each of the $K - 1$ terms of the previous summation.

The combination of the two terms in the first summation requires $K - 1$ additions while the summation itself requires $K - 2$ additions. The combination of the expectation terms requires 1 multiplication of the constant 2 and $\frac{1}{2}(K - 1)(K - 2) - 1$ additions. The last term with the noise estimates requires 1 multiplication for the constant 2, $\frac{1}{2}(K - 1)(K - 2)$ multiplications of the two noise estimates, and $\frac{1}{2}(K - 1)(K - 2) - 1$ additions for the summation of these values. These last two terms are only computed when $K > 2$. The second term requires 1 addition, 1 subtraction, and 1 multiplication to combine these terms together.

The combination of the first and second terms require 1 additional addition. When $K = 2$, the calculation of each noise variance requires only 4 calculations for each user. However, for $K > 2$, the computation of each noise variance requires:

\[
(K - 2 + 1)(K - 1) + (K - 1) + (K - 2) + 1 + \frac{1}{2}(K - 1)(K - 2) - 1 + 1 + \frac{1}{2}(K - 1)(K - 2) - 1 + 1 + 1 + 1 = 5 + \frac{5}{2}K^2 - \frac{9}{2}K + 5
\]
calculations for each user per input bit.

The calculation of each variance in (A.11) requires $4K - 4 + 1 + 4 = 4K + 1 = 4(2) + 1 = 9$ calculations for each user when $K = 2$. For $K > 2$, the calculation of each noise variance requires:

\[
4K - 4 + 1 + \frac{5}{2}K^2 - \frac{9}{2}K + 5 = \frac{5}{2}K^2 - \frac{1}{2}K + 2
\]
calculations for each user per input bit.

The coded bit variance $v_i^2(k)$ used in the calculation of the noise variance is computed from (3.20) as $1 - \left[ P_r \left\{ d_t^{(k)} = 1 \right\} - P_r \left\{ d_t^{(k)} = -1 \right\} \right]^2$. This term requires 2 subtractions and 1 multiplication for each user. Therefore, $3K$ total calculations are required to compute and store these values for all $K$ users per input bit.

There are a total of $\frac{1}{2} K(K-1)$, $K \geq 3$, expectation values $E \left\{ \tilde{n}_t^{(i)} \tilde{n}_t^{(j)} \right\}$ which are pre-calculated and stored by the metric generator and are computed through the equation derived in (3.47). This equation is comprised of eight separate terms that require 2 subtractions and 5 additions to combine them together. The constants $(\rho^2 + \rho)$ and $(\rho^2 + 1)$ are pre-calculated and stored, as is $2\rho^2$. The squares of the coded bit means $m_i^2(k)$ are also pre-calculated for all $K$ users and stored, requiring $K$ multiplications.

The first of these terms needs 1 multiplication. The second and third terms each require $K - 2$ additions for the summation, 1 multiplication for $\rho$, 1 addition to combine the terms, and 1 multiplication to finish the combination of the term. The fourth and fifth term each require $K - 3$ additions and 1 multiplication for the summation term, 2 additions to combine the coded bit variances, and 1 multiplication for $\rho$ to complete this term.

The sixth term of (3.47) requires $K - 3$ additions and $K - 2$ multiplications for each of the summation terms. This term also requires 1 addition and 1 multiplication for the constant $(\rho^2 + \rho)$ to complete this term. The seventh term needs 2 multiplications. The eighth and last term requires the addition of $\frac{1}{2}(K-2)(K-3)$ terms with 1 multiplication per term. The summation of these terms requires
\( \frac{1}{2}(K-2)(K-3) - 1 \) additions. The multiplication of the constant \( 2\rho^2 \) requires 1 multiplication. This last term is not computed when \( K = 3 \).

Adding these calculations together, each expectation \( E\{n_t^{(i)} \bar{n}_t^{(j)}\} \) requires \( K^2 + 3K + 12 \) calculations when \( K > 3 \). Only 29 calculations are required for each expectation term when \( K = 3 \).

### A.4 Overall Decoder Complexity

The previous sections divided the iterative multi-user CDMA decoder into its smaller components. In this section, the overall complexity of the various iterative decoders is found. The overall complexity of the iterative decoder is dependent on the number of iterations being used in the system. As in chapter 4, the multi-user iterations are referred to as “MUit”, the Turbo Code decoder iterations are referred to as “TDit”, and “MAIt” refers to the iterations in which only the MAI distribution is updated in the correlated Gaussian noise models.

#### Single-User Complexity

The complexity of the single-user system utilizing Turbo Codes for FEC consists of the complexity of the Turbo Codes and the calculation of the Gaussian distribution for the systematic and parity bits. The Turbo Code decoders for the single-user system are not required to compute the LLR values for the parity bits or convert the LLR values to bit probabilities. This reduced complexity Turbo Code decoder requires \( 26(2^{P+1}) + 8 \) calculations per iteration of the Turbo decoder.

The additive Gaussian noise in the single-user system has zero mean and vari-
APPENDIX A. COMPLEXITY CALCULATIONS

The calculation of the Gaussian distribution, given in (A.6), can be simplified to:

\[ p\left( r_i^{(k)} | d_i^{(k)} = d \right) = \frac{1}{\sqrt{\pi N_0}} \exp \left[ -\frac{(r_i^{(k)} - d)^2}{N_0} \right], \quad (A.12) \]

where \( d \in \{-1, 1\} \). The constant term \( 1/\sqrt{\pi N_0} \) can be omitted without affecting the BJCR algorithm. The exponential constant \( e^{N_0} \) is assumed to be pre-calculated and stored. The calculation of the Gaussian distribution only requires 1 subtraction and 1 multiplication for the numerator with 1 multiplication for the exponential constant. This distribution needs to be computed for both symbol values \( d \in \{-1, 1\} \) and for each of the systematic and parity bits. Thus, \( 2(3)(3) = 18 \) calculations are required to compute the Gaussian probabilities corresponding to one input bit.

The calculation of the Gaussian probability distributions are considered overhead since they are calculated and stored prior to the start of the Turbo decoding algorithm. The final hard decision, requiring 1 calculation, is also considered overhead. The complexity of the single-user system is:

\[ [26(2^{P+1}) + 8] TDit + 19 \quad (A.13) \]

total calculations per input bit.

Continuous Gaussian Approximation of MAI Complexity

The generation of the mean and variances of the systematic and parity bits for the continuous Gaussian approximation of the MAI decoder is found to require
APPENDIX A. COMPLEXITY CALCULATIONS

18K − 12 calculations per user. Another 30 calculations are required to generate the three probability metrics for each input bit per user. Therefore, the metric generator requires \((18K - 12 + 30)K = 18K^2 + 18K\) total calculations to compute the probability metrics for all \(K\) users corresponding to one input bit. These calculations are performed for each multi-user iteration.

The \(K\) individual Turbo Code decoders in the multi-user decoder each require \(36(2^{P+1}) + 18\) calculations for each iteration of the Turbo Code decoders. These calculations are also required for each multi-user iteration. The final hard decision is made only at the very end of the iterative process and adds 1 calculation per user to the complexity of the decoder.

Adding all of these calculations together results in the complexity of the continuous Gaussian approximation of the MAI decoder being:

\[
[18K^2 + 18K] \text{MUit} + [(36(2^{P+1})K + 18K) \text{T Dit}] \text{MUit} + K
\]  
(A.14)
total calculations per input bit for all \(K\) users, \(K \geq 2\).

Discrete Analysis of MAI Complexity

The metric generator for the discrete analysis of the MAI decoder computes a length \(K\) discrete sequence \(\{P_{t,m}^{(k)}\}\) for each of the \(K\) users requiring the total number of computations found from (A.8) and (A.9) for each of the metrics for the systematic and parity bits when \(K > 2\). If \(K = 2\), no calculations are needed to compute the convolution since only one user is providing the interference. The computation of the three metrics for each user’s input bit was also found to require \(36K\) calculations.
per user. These calculations are required for each multi-user iteration.

Combining these calculations with those required by the Turbo Code decoder and the final hard decision, the overall complexity of the discrete analysis of the MAI decoder when \( K \geq 3 \) is:

\[
\left[36K^2 + 2 \sum_{m=1}^{\left\lfloor \frac{K-1}{2} \right\rfloor} \left( \frac{K-1}{K-m} \right)(K-2) + c \left( \frac{K-1}{K - \left\lceil \frac{K}{2} \right\rceil} \right)(K-2) \right] 3K \\
+ \left( 2 \sum_{m=1}^{\left\lfloor \frac{K-1}{2} \right\rfloor} \left[ \left( \frac{K-1}{K-m} \right) - 1 \right] + c \left[ \left( \frac{K-1}{K - \left\lceil \frac{K}{2} \right\rceil} \right) - 1 \right] \right) 3K \right] \text{MUit} \\
+ \left[ (36(2^P+1)K + 18K) \text{T Dit} \right] \text{MUit} + K
\]

(A.15)

total calculations for each input bit of all \( K \) users, where

\[
c = \begin{cases} 
1 & : \ K \ \text{odd} \\
2 & : \ K \ \text{even}.
\end{cases}
\]

When \( K = 2 \), the complexity of the discrete analysis of the MAI decoder is:

\[
[144] \text{MUit} + \left[ (36(2^P+1)K + 18K) \text{T Dit} \right] \text{MUit} + K. \quad (A.16)
\]

**Correlated Gaussian Noise Model Complexity**

The correlated Gaussian noise model is only used after a set number of iterations of the multi-user decoder updating only the probability distribution of the MAI using the continuous Gaussian approximation for the MAI. Thus, the complexity of the decoder using the correlated Gaussian noise model will depend on the values of the
three different iterations introduced earlier.

The first MAI\textsubscript{it} iterations will have the same complexity as found for the continuous Gaussian approximation decoder. The subsequent complexity will include the complexity added to the decoder by updating the Gaussian noise probability distribution. The calculation of the mean and variances for the three metrics required for each user is found to need $21K - 12$ calculations while $30$ calculations are required for each user to compute the value of these metrics. The computation of the noise estimates required $3K^2 + 6K$ total calculations for all $K$ users, independent of whether hard or soft decisions are made on the coded bit probabilities.

The overall complexity of the iterative decoder utilizing the correlated Gaussian model, for both hard and soft decisions of the coded bit probabilities, is found to be:

\[
\left[18K^2 + 18K\right] \text{MAI}_{\text{it}} + \left[\left(36(2^P + 1)K + 18K\right)T\text{D}_{\text{it}}\right] \text{MAI}_{\text{it}} \\
+ \left[24K^2 + 24K\right] \text{MU}_{\text{it}} + \left[\left(36(2^P + 1)K + 18K\right)T\text{D}_{\text{it}}\right] \text{MU}_{\text{it}} + K
\] (A.17)

total calculations per input bit for all $K$ users, $K \geq 2$.

**Extended Correlated Gaussian Noise Model Complexity**

As in the correlated Gaussian noise model, the extended correlated Gaussian noise model is only used after a set number of iterations updating the probability distribution of the MAI only. Since the continuous Gaussian approximation is used to update the MAI, the first number of iterations will have the same complexity found for this decoder.
APPENDIX A. COMPLEXITY CALCULATIONS

The iterations utilizing the extended correlated Gaussian noise model will have the complexity of all the components analyzed previously. The combination of all these calculations for all $K$ users and the three metrics of the systematic and parity bits results in the overall complexity of the extended correlated Gaussian noise model decoder being for $K = 2$:

$$[108] \text{MAI}_t + \left[ \left( 72(2^{P+1}) + 36 \right) \text{T Dit} \right] \text{MAI}_t$$
$$+ [186] \text{MU}_t + \left[ \left( 72(2^{P+1}) + 36 \right) \text{T Dit} \right] \text{MU}_t + 2 \quad (A.18)$$

for $K = 3$:

$$[216] \text{MAI}_t + \left[ \left( 108(2^{P+1}) + 54 \right) \text{T Dit} \right] \text{MAI}_t$$
$$+ [711] \text{MU}_t + \left[ \left( 108(2^{P+1}) + 54 \right) \text{T Dit} \right] \text{MU}_t + 3 \quad (A.19)$$

and for $K > 3$:

$$\left[ 18K^2 + 18K \right] \text{MAI}_t + \left[ \left( 36(2^{P+1})K + 18K \right) \text{T Dit} \right] \text{MAI}_t$$
$$+ \frac{1}{2} \left[ 3K^4 + 21K^3 + 51K^2 + 63K \right] \text{MU}_t$$
$$+ \left[ \left( 36(2^{P+1})K + 18K \right) \text{T Dit} \right] \text{MU}_t + K \quad (A.20)$$

total calculations for each input bit of all $K$ users.
Bibliography


