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

Dwayne Stienstra and A. K. Khandani

Department of Electrical and Computer Engineering

University of Waterloo

Waterloo, Ontario, Canada, N2L 3G1

Technical Report UW-E&CE#2000-4

March 27, 2000

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 report, 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.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

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 1.1, 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. Code-Division Multiple Access (CDMA)[1] systems are one such technique that much recent research has focused on. It 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. 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]. This is the multiple access technique used in the communication system of this report.

[1] 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 report will only focus on DS-CDMA and, thus, CDMA will be used in place of DS-CDMA henceforth.

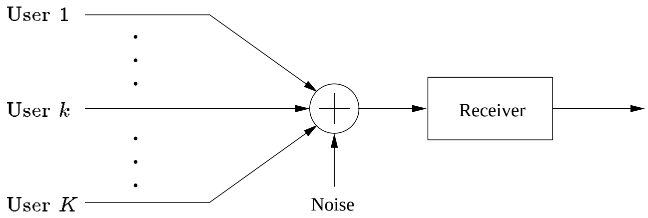

Figure 1.1: Multiple Access Communication System

In a CDMA system[2], each user is assigned a unique noiselike spreading signal that is used to spread the user information sequence over the assigned frequency band, using more bandwidth than what is necessary to transmit the information. The spread information signal is then transmitted over the shared channel. 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 interference among users is small. 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 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[3], 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

[2] Only a brief description of a CDMA system is presented here. For a more detailed description, see references [5] and [7].

[3] A bank of correlators in place of the matched filters can equivalently be utilized to implement this receiver.

users is small but deteriorates rapidly as the number of users increases.

For non-orthogonal code signals, CDMA performance can be greatly enhanced by jointly decoding 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 derivation of the optimum multi-user detector was based on a maximum likelihood sequence detection formulation for asynchronous users transmitting over an additive white Gaussian noise (AWGN) channel. 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. Duel-Hallen *et al.* [15] and the references within provide a further description of most of these multi-user receivers.

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 [12-14,16,17] incorporate an FEC code in the design of the decoders, resulting in improved BER performances at lower SNR values.

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[4]. The Turbo Code technique combines the concepts of iterative decoding, soft-in/soft-out decoding, and non-uniform random interleaving. The corresponding decoding algorithm is a modification of the BCJR [2]

[4] Claude Shannon presented theoretical limits of the performance of a communication system in his 1948 paper [6].

algorithm applied to the decoding of Recursive Systematic Convolutional (RSC) codes. Berrou *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 report since they perform much better than other channel coding techniques.

Gaillorenzi and Wilson [16] 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.

Reed *et al.* [12, 13, 14] proposed a sub-optimal iterative multi-user decoder 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. This work is extended to the asynchronous CDMA system in [17].

This report 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 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 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 report 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 report is organized as follows: Chapter 2 examines the CDMA system model used in the development of the proposed iterative multi-user decoders and gives a brief description of Turbo Codes. Chapter 3 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.

Chapter 2

Direct-Sequence CDMA System Model

2.1 Introduction

The system model used throughout this report 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, i.e., all user signals are received at the same power level, and that the channel is both chip and symbol synchronous[1]. This chapter examines how these assumptions affect this model.

2.2 System Description

The basic continuous-time K user CDMA channel model used in this report consists of a sum of modulated synchronous signature waveforms transmitted over an additive white Gaussian noise (AWGN) channel, each user transmitting M source bits. The chip and symbol synchronous,

[1] Note that the generalization to an asynchronous channel is straightforward if the MAI term is treated properly.

continuous-time CDMA signal for one source bit at the receiver is represented as[2]:

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

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 k^{th} user coded bit $d_k \in \{-1, +1\}$ and the white Gaussian noise $n(t)$ with unit power spectral density added to the received signal, where the one-sided spectral level is $N_0 = 2\sigma^2$.

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:

$$\rho_t^{(ij)} = \int_0^T s_i(t)s_j(t)dt. \tag{2.2}$$

The cross-correlation values may be time-varying if random spreading signals are utilized. The cross-correlation values can be collected into a cross-correlation matrix $\mathbf{H}_t = \{\rho_t^{(ij)}\}$, which has diagonal elements equal to 1 and is symmetric nonnegative definite[3].

The front-end of the receiver for this report uses K coherently matched filters[4], each matched to its corresponding user spreading signal $s_k(t)$, the output of which is sampled at rate T to convert the continuous-time signal $r(t)$ into a discrete-time signal $\mathbf{r}_t$. 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

[2] Throughout this report, the notations follow the standard: scalars are lower case, vectors are highlighted and lower case, and matrices are highlighted and upper case.

[3] The cross-correlation matrix $\mathbf{H}_t$ is positive definite if and only if the spreading signals $s_k(t)$, $k \in \{1, \ldots, K\}$, are linearly independent

[4] Note that a RAKE receiver with a few modifications could also be used for the front-end of this receiver.

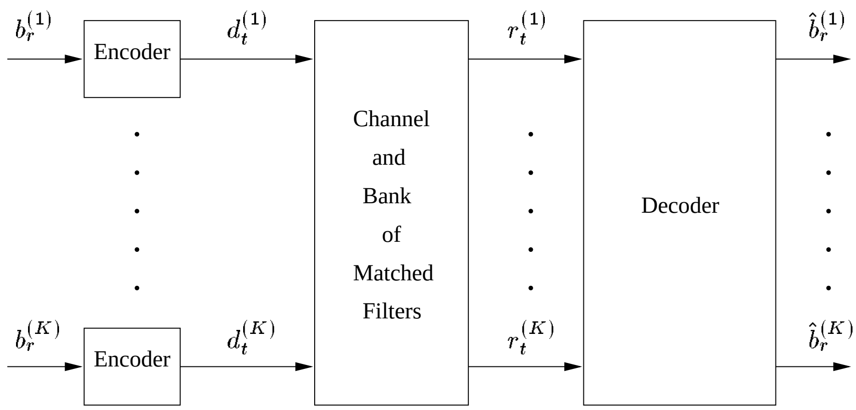

Figure 2.1: Discrete-Time CDMA System Model

report to reduce the complexity of the equations.

The block diagram of the equivalent discrete-time CDMA system is shown in Figure 2.1. 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[5] of the block of data, and R is the coding rate. The coded symbols are 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 $\mathbf{s}_t^{(k)} \in \left\{\frac{-1}{\sqrt{N}}, \frac{1}{\sqrt{N}}\right\}^N$. 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

$$r_t^{(k)} = d_t^{(k)} + \sum_{\substack{i=1 \\ i \neq k}}^{K} d_t^{(i)} \rho_t^{(ik)} + n_t^{(k)}, \tag{2.3}$$

where $n_t^{(k)}$ is a Gaussian random process with zero mean and variance $\sigma^2 = N_0/2$ [11].

[5]Note that the time variable t is used in both the continuous-time and discrete-time models. For the remainder of this report t is used as a discrete variable.

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

$$\mathbf{r}_t = \mathbf{H}_t \mathbf{d}_t + \mathbf{n}_t, \tag{2.4}$$

where[6] $\mathbf{r}_t = \left(r_t^{(1)}, \ldots, r_t^{(K)}\right)^\top$ 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 auto-correlation 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 report.

2.3 Turbo Codes

Each user utilizes Turbo Codes for FEC channel coding in the iterative multi-user receiver. The iterative decoder utilizes the the individual Turbo Codes in combination with the iterative process to reduce the Bit Error Rate (BER) performance of each user. This section briefly introduces the Turbo Code encoder and decoder.

Figure 2.2 illustrates a two-code Turbo Code encoder consisting of a parallel concatenation of two identical RSC component codes, each with $P = 4$ memory elements. The RSC component code with generator polynomials $\mathbf{g}_1 = 37_8$ and $\mathbf{g}_2 = 21_8$ (denoted in octal base) is used for both RSC encoders in this report, each assumed to start and end in the all zero state.

The first RSC encoder uses the k^{th} users source bit stream $\left\{b_r^{(k)}\right\}$ to generate $\left\{X_r^{(k)}\right\}$, the systematic bits generated by replicating the source bits $b_r^{(k)}$ and $\left\{Y_{1,r}^{(k)}\right\}$, 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. For the second RSC encoder, the k^{th} user's source bit stream $\left\{b_r^{(k)}\right\}$ is passed through a pseudo-random interleaver to form the second RSC encoder input sequence $\left\{b_\nu^{(k)}\right\}$, $\nu \in \{1, \ldots, M\}$. Here, the $b_r^{(k)}$ and $b_\nu^{(k)}$ are related by $\nu = INT(r)$, meaning that the r^{th} bit in the original data sequence has been moved to the ν^{th} position in the interleaved sequence. The second binary RSC encoder uses this pseudo-random interleaved source

[6] The symbol $(\cdot)^\top$ represents the transposition operation and is used throughout this report.

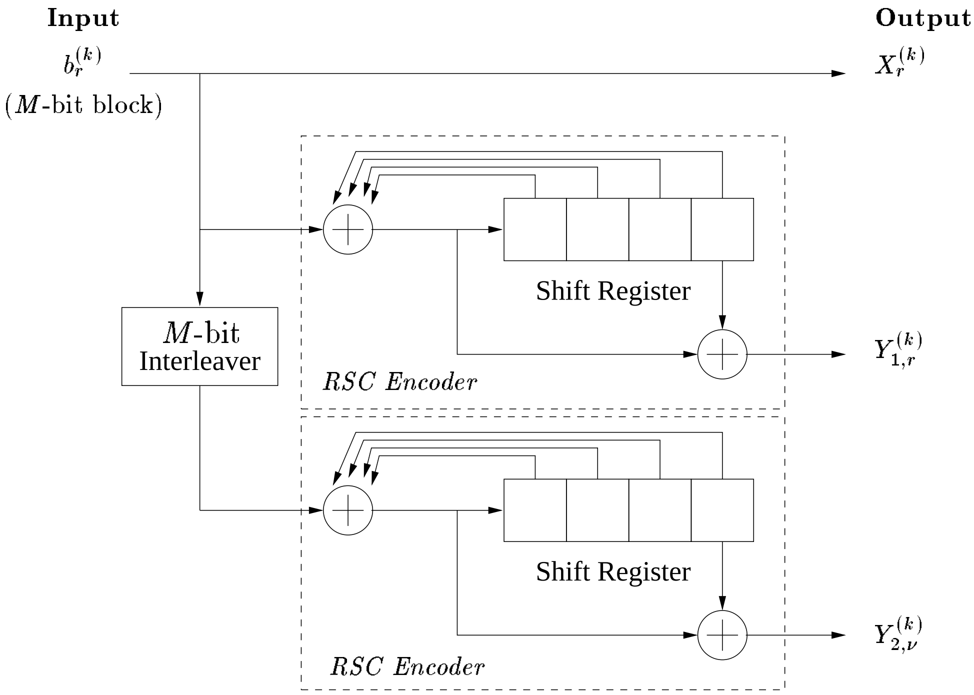

Figure 2.2: Rate 1/3 Turbo Code Encoder

bit sequence to generate the parity bit sequence $\left\{Y_{2,\nu}^{(k)}\right\}$.

The parallel concatenation of the two RSC encoders produces a rate $R = 1/3$ code. The resulting bit stream is Binary Phase-Shift Key (BPSK) modulated to produce the length L transmitted sequence:

$$\left\{d_t^{(k)}\right\} = \left\{\left(2X_r^{(k)} - 1, 2Y_{1,r}^{(k)} - 1, 2Y_{2,\nu}^{(k)} - 1\right)\right\}, \qquad t \in \{1, \ldots, L\}. \tag{2.5}$$

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

The received sequence of length L for user k can be expressed in terms of the systematic and parity bits as $\{r_t^{(k)}\} = \{(x_r^{(k)}, y_{1,r}^{(k)}, y_{2,\nu}^{(k)})\}$, $t \in \{1, \ldots, L\}$, $r, \nu \in \{1, \ldots, M\}$, where from (2.3):

$$x_r^{(k)} = \left(2X_r^{(k)} - 1\right) + \sum_{\substack{i=1 \\ i \neq k}}^{K} \left(2X_r^{(i)} - 1\right) \rho_r^{(ik)} + \zeta_r^{(k)} \tag{2.6}$$

$$y_{1,r}^{(k)} = \left(2Y_{1,r}^{(k)} - 1\right) + \sum_{\substack{i=1 \\ i \neq k}}^{K} \left(2Y_{1,r}^{(i)} - 1\right) \rho_{1,r}^{(ik)} + \eta_r^{(k)} \tag{2.7}$$

$$y_{2,\nu}^{(k)} = \left(2Y_{2,\nu}^{(k)} - 1\right) + \sum_{\substack{i=1 \\ i \neq k}}^{K} \left(2Y_{2,\nu}^{(i)} - 1\right) \rho_{2,\nu}^{(ik)} + \xi_\nu^{(k)}, \tag{2.8}$$

$\zeta_r^{(k)}$, $\eta_r^{(k)}$, 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.

The Turbo Code decoder consists of two RSC decoders. 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. Berrou *et al.* [1] applied a variation of the BCJR algorithm for the decoding of RSC encoders. Robertson [3] and Jung [4] made further improvements to reduce the complexity and memory storage of the algorithm. A brief description of the modified BCJR algorithm and the modifications proposed in [3] is presented in this report. 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 decription as an identical algorithm will be used in the Turbo Code decoder of each user.

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 RSC decoder uses R_1^M to calculate the LLR value for each bit. The LLR is given by the equation:

$$\Lambda(b_r) = \ln \left(\frac{P_r\left\{b_r = 1 | R_1^M\right\}}{P_r\left\{b_r = 0 | R_1^M\right\}} \right), \tag{2.9}$$

where $P_r\left\{b_r = i | R_1^M\right\}$ 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 $\beta_r(m)$.

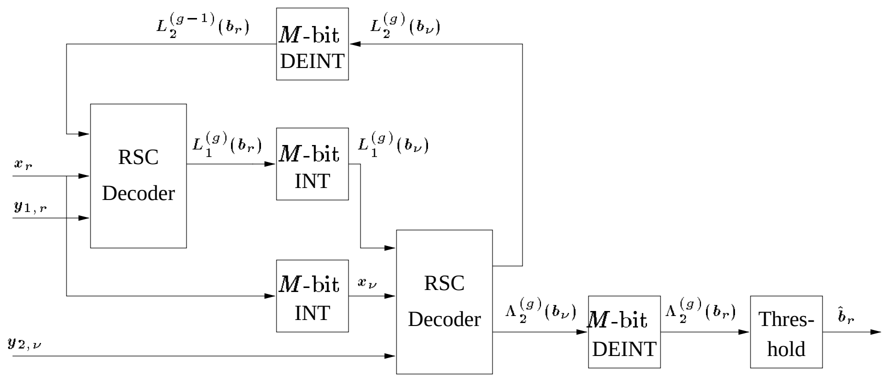

Figure 2.3: Two-Code Turbo Code Decoder

The Turbo Code decoder suggested by Robertson [3] uses this algorithm in an iterative manner with two RSC decoders and is shown in Figure 2.3 for the g^{th} iteration. 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 $\{R_r^1\} = \{R_1^1, \ldots, R_M^1\}$, where $R_r^1 = (x_r, y_{1,r})$, while the second component decoder receives the sequence $\{R_\nu^2\} = \left\{R_{INT(1)}^2, \ldots, R_{INT(M)}^2\right\}$, where $R_\nu^2 = (x_\nu, y_{2,\nu})$. 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, $y_{1,r}$, and $y_{2,\nu}$ were previously given in (2.6), (2.7), and (2.8) respectively for user k.

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:

$$\Lambda_1^{(g)}(b_r) = I_r + L_2^{(g-1)}(b_r) + L_1^{(g)}(b_r). \tag{2.10}$$

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[7] information I_r do not need to be passed to the second decoder.

[7] 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 second RSC decoder uses the received sequence $\{R_\nu^2\}$ 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. The LLR $\Lambda_2^{(g)}(b_\nu)$ for the second decoder can also be factored as:

$$\Lambda_2^{(g)}(b_\nu) \quad = \quad I_\nu + L_1^{(g)}(b_\nu) + L_2^{(g)}(b_\nu)\,. \tag{2.11}$$

The extrinsic information $L_1^{(g)}(b_r)$ and $L_2^{(g)}(b_\nu)$ are also considered to be Log-Likelihood Ratios. The extrinsic information $L_2^{(g)}(b_\nu)$ is de-interleaved and passed to the first decoder for the next iteration.

The extrinsic information $L_2^{(0)}(b_r)$ 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.

Only after a set number of iterations have been completed are the LLR values $\Lambda_2^{(g)}(b_\nu)$ compared to get an estimate of the bit b_r transmitted. The LLR values $\Lambda_2^{(g)}(b_\nu)$ are first de-interleaved to $\Lambda_2^{(g)}(b_r)$. A final hard-decision is made by comparing $\Lambda_2^{(g)}(b_r)$ to a threshold of zero. Thus,

$$\hat{b}_r = \begin{cases} 1 & : \quad \Lambda_2^{(g)}(b_r) > 0 \\ 0 & : \quad \Lambda_2^{(g)}(b_r) \leq 0 \end{cases} \tag{2.12}$$

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

Chapter 3

Iterative Multi-User CDMA Receiver

3.1 Iterating The CDMA Decoder

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 (2.3). 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 proposed iterative multi-user CDMA decoder is illustrated in Figure 3.1. The decoder is composed of a metric generator and K individual FEC decoders. The metric generator uses the outputs of the matched filters, $\mathbf{r}_t = \left(r_t^{(1)}, \ldots, r_t^{(K)}\right)^\top$, 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 iteration. 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 gener-

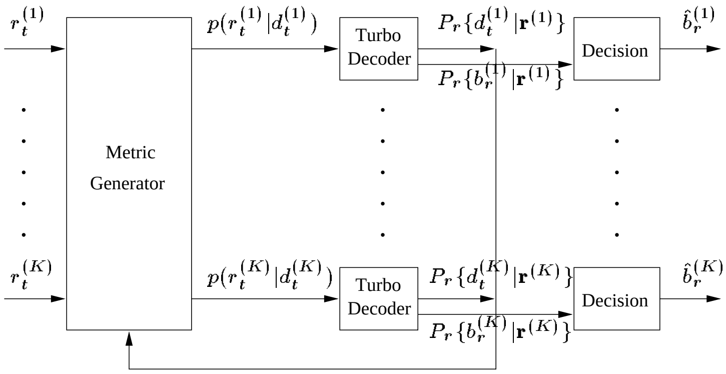

Figure 3.1: Iterative Multi-User Decoder

ating the correct probability information in the metric generator for use in the FEC decoders. The key point to the proposed iterative algorithm is to find a proper method to update the corresponding marginal probability distribution, $p\left(r_t^{(k)}|d_t^{(k)}\right)$, from iteration to iteration. This report explores different methods to extract and update these marginal probabilities.

For the iterative multi-user CDMA receiver, each Turbo Code decoder is modified to produce both uncoded and coded bit probabilities $P_r\left\{b_r^{(k)}=b|\mathbf{r}^{(k)}\right\}$, $b\in\{0,1\}$, and $P_r\left\{d_t^{(k)}=d|\mathbf{r}^{(k)}\right\}$, $d\in\{-1,1\}$, 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.

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 *a-priori* input probability for the $i+1$ iteration in order to form the proper probability distribution. For the initial iteration, the metric generator assumes equally likely bit probabilities $P_r\left\{d_t^{(k)}=-1\right\}$ and $P_r\left\{d_t^{(k)}=1\right\}$ for all K users. 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 *a-priori* bit probabilities of the next iteration through the equation $P_r\left\{d_t^{(k)}=d\right\} = P_r\left\{d_t^{(k)}=d|\mathbf{r}^{(k)}\right\}$, $d\in\{-1,+1\}$. After the required iterations have been completed, the systematic bit probabilities $P_r\left\{b_r^{(k)}=b|\mathbf{r}^{(k)}\right\}$, $b\in\{0,1\}$, generated by the K Turbo Code decoders are used to make the final hard-decision to calculate $\hat{b}_r$, as in (2.12).

To reduce the complexity of the derivations for the probability metric, the cross-correlation 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. As a result of this assumption, the discrete cross-correlation matrix $\mathbf{H}_t$ becomes a time invariant matrix, represented as $\mathbf{H}$, which has diagonals equal to one and the non-diagonal elements equal to ρ.

3.1.1 Continuous Gaussian Approximation of MAI

From (2.3), the MAI term consists of $K-1$ discrete binary random variables. From the central limit theorem[1], 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 report, the MAI term is approximated by a Gaussian random variable.

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

$$\begin{aligned} m_t^{(k)} &= E\left\{d_t^{(k)}\right\} = \sum_{d \in \{-1,1\}} d P_r\left\{d_t^{(k)}=d\right\} \\ &= P_r\left\{d_t^{(k)}=1\right\} - P_r\left\{d_t^{(k)}=-1\right\} \qquad (3.1) \\ v_t^{2\,(k)} &= E\left\{d_t^{2\,(k)}\right\} - E\left\{d_t^{(k)}\right\}^2 \\ &= 1 - \left[P_r\left\{d_t^{(k)}=1\right\} - P_r\left\{d_t^{(k)}=-1\right\}\right]^2. \qquad (3.2) \end{aligned}$$

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

$$\mu_{t,\,MAI}^{(k)} = \rho \sum_{\substack{i=1 \\ i \neq k}}^{K} \left[P_r\left\{d_t^{(i)}=1\right\} - P_r\left\{d_t^{(i)}=-1\right\}\right] = \rho \sum_{\substack{i=1 \\ i \neq k}}^{K} m_t^{(i)} \qquad (3.3)$$

[1] The central limit theorem states that the sum of n statistically independent random variables approach a Gaussian distribution as $n \to \infty$.

$$\sigma_{t,\,MAI}^{2\,(k)} = \rho^2 \sum_{\substack{i=1 \\ i\neq k}}^{K} \left(1 - \left[P_r\left\{d_t^{(i)} = 1\right\} - P_r\left\{d_t^{(i)} = -1\right\}\right]^2\right) = \rho^2 \sum_{\substack{i=1 \\ i\neq k}}^{K} v_t^{2\,(k)}. \qquad (3.4)$$

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:

$$\mu_t^{(k)} = d + \mu_{t,\,MAI}^{(k)} \qquad (3.5)$$

$$\sigma_t^{2\,(k)} = \sigma^2 + \sigma_{t,\,MAI}^{2\,(k)}. \qquad (3.6)$$

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 following sections.

3.1.2 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 $\rho \sum_{\substack{i=1 \\ i\neq k}}^{K} d_t^{(i)}$ 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[2] the other $K - 1$ sets of coded bit probabilities to produce the length K discrete probability sequence:

$$\begin{aligned} \left\{P_{t,m}^{(k)}\right\} &= \left\{P_r\left\{\sum_{\substack{i=1 \\ i\neq k}}^{K} d_t^{(i)} = m\right\}\right\} \\ &= \left\{P_r\left\{d_t^{(1)} = -1\right\}, P_r\left\{d_t^{(1)} = 1\right\}\right\} * \ldots \end{aligned}$$

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

$$\begin{aligned}&*\left\{P_r\left\{d_t^{(k-1)}=-1\right\},P_r\left\{d_t^{(k-1)}=1\right\}\right\}\\&*\left\{P_r\left\{d_t^{(k+1)}=-1\right\},P_r\left\{d_t^{(k+1)}=1\right\}\right\}*\ldots\\&*\left\{P_r\left\{d_t^{(K)}=-1\right\},P_r\left\{d_t^{(K)}=1\right\}\right\},\end{aligned}\tag{3.7}$$

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

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 σ^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:

$$p\left(r_t^{(k)}|d_t^{(k)}=d\right)=\sum_m \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{\left(r_t^{(k)}-d-\rho m\right)^2}{2\sigma^2}\right]\times P_{t,m}^{(k)},\tag{3.8}$$

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

3.1.3 Correlated Gaussian Noise Model

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 by the metric generator through the use of the coded bit probabilities at the output of the Turbo Code decoders from the previous iteration.

Upon generating an estimate of the noise samples, the metric generator calculates the conditional distribution $p\left(n_t^{(k)}|n_t^{(1)}, \overset{\cdots}{\neq k}, n_t^{(K)}\right)$ to replace the noise distribution $p\left(n_t^{(k)}\right)$ in the calculation of the metric $p\left(r_t^{(k)}|d_t^{(k)}\right)$. 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

$p\left(n_t^{(k)}|n_t^{(1)}, \underset{\neq k}{\dots}, n_t^{(K)}\right)$ 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:

$$p\left(n_t^{(k)}|n_t^{(1)}, \underset{\neq k}{\dots}, n_t^{(K)}\right) = \frac{p\left(n_t^{(1)}, \dots, n_t^{(k)}, \dots, n_t^{(K)}\right)}{p\left(n_t^{(1)}, \underset{\neq k}{\dots}, n_t^{(K)}\right)}. \tag{3.9}$$

The noise vector $\mathbf{n}_t = \left(n_t^{(1)}, \dots, n_t^{(K)}\right)^\top$ has the probability distribution:

$$\begin{aligned} p(\mathbf{n}_t) &= p\left(n_t^{(1)}, \dots, n_t^{(k)}, \dots, n_t^{(K)}\right) \\ &= \frac{1}{(2\pi)^{K/2}|\mathbf{H}|^{1/2}} \exp\left[-\frac{1}{2}\mathbf{n}_t^\top \mathbf{H}^{-1}\mathbf{n}_t\right]. \end{aligned} \tag{3.10}$$

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

Expanding and simplifying (3.9), the conditional noise probability distribution takes the form:

$$p\left(n_t^{(k)}|n_t^{(1)}, \underset{\neq k}{\dots}, n_t^{(K)}\right) = \frac{1}{\sqrt{2\pi}\tilde{\sigma}_t^{(k)}} \exp\left[-\frac{\left(n_t^{(k)} - \tilde{\mu}_t^{(k)}\right)^2}{2\tilde{\sigma}_t^{2\,(k)}}\right], \tag{3.11}$$

which is a Gaussian random variable with mean and variance:

$$\tilde{\mu}_t^{(k)} = \frac{\rho}{(K-2)\rho+1} \sum_{\substack{i=1 \\ i\neq k}}^{K} n_t^{(i)} \tag{3.12}$$

$$\tilde{\sigma}_t^{2\,(k)} = \tilde{\sigma}^2 = \sigma^2\left[\frac{(1-\rho)(1+(K-1)\rho)}{(1+(K-2)\rho)}\right], \tag{3.13}$$

respectively. It is observed that the conditional noise variance $\tilde{\sigma}^2$ is constant while 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 $\mu_t^{(k)} = d + \mu_{t,\,MAI}^{(k)} + \tilde{\mu}_t^{(k)}$ and variance $\sigma_t^{2\,(k)} = \sigma_{t,\,MAI}^{2\,(k)} + \tilde{\sigma}^2$, where $\mu_{t,\,MAI}^{(k)}$ and $\sigma_{t,\,MAI}^{2\,(k)}$ are the MAI Gaussian distribution's mean and variance given in (3.3) and (3.4).

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:

$$d_t^{(k)} = \begin{cases} -1 & : \quad P_r\left\{d_t^{(k)}=-1\right\} \geq P_r\left\{d_t^{(k)}=1\right\} \\ 1 & : \quad P_r\left\{d_t^{(k)}=-1\right\} < P_r\left\{d_t^{(k)}=1\right\}. \end{cases} \tag{3.14}$$

The estimated coded bits are used to produce an estimate of the noise for user k at time t through the equation:

$$n_t^{(k)} = r_t^{(k)} - d_t^{(k)} - \rho \sum_{\substack{i=1 \\ i \neq k}}^{K} d_t^{(i)}. \tag{3.15}$$

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 (3.1) as:

$$m_t^{(k)} = E\left\{d_t^{(k)}\right\} = P_r\left\{d_t^{(k)} = 1\right\} - P_r\left\{d_t^{(k)} = -1\right\}, \quad k \in \{1, \ldots, K\}. \tag{3.16}$$

These average values of the coded bits are used to generate a soft estimate of the noise through an equation similar to (3.15) as:

$$\hat{n}_t^{(k)} = r_t^{(k)} - m_t^{(k)} - \rho \sum_{\substack{i=1 \\ i \neq k}}^{K} m_t^{(i)}. \tag{3.17}$$

3.1.4 Extending The Correlated Gaussian Noise Model

The estimates of the noise derived in the previous section are not exact but are random variables. As such, the noise estimate $\bar{n}_t^{(k)}$ can be approximated by a Gaussian distribution:

$$\bar{n}_t^{(k)} \sim N\left(\hat{n}_t^{(k)}, v_t^{2\ (k)} + \rho^2 \sum_{\substack{i=1 \\ i \neq k}}^{K} v_t^{2\ (i)}\right), \tag{3.18}$$

where $v_t^{2\ (k)}$ is the variance of the coded bits given in (3.2) and $\hat{n}_t^{(k)}$ is given by (3.17), independent of 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 $p\left(n_t^{(k)}\right)$ for user k is obtained by integrating the product of the conditional probability distribution (3.11) and the joint probability distribution $p\left(\bar{n}_t^{(1)}, \underset{\neq k}{\cdots}, \bar{n}_t^{(K)}\right)$ to yield:

$$\begin{aligned} p\left(n_t^{(k)}\right) &= \int_{\bar{n}_t^{(1)}} \underset{\neq k}{\cdots} \int_{\bar{n}_t^{(K)}} p\left(\bar{n}_t^{(1)}, \underset{\neq k}{\cdots}, \bar{n}_t^{(K)}\right) \times \\ &\qquad p\left(n_t^{(k)} | n_t^{(1)} = \bar{n}_t^{(1)}, \underset{\neq k}{\cdots}, n_t^{(K)} = \bar{n}_t^{(K)}\right) d\bar{n}_t^{(1)} \underset{\neq k}{\cdots} d\bar{n}_t^{(K)}. \end{aligned} \tag{3.19}$$

Since both the conditional and joint probabilities distributions are Gaussian, the probability distribution $p\left(n_t^{(k)}\right)$ will also be Gaussian with mean and variance[3]:

$$\bar{\mu}_t^{(k)} = \frac{\rho}{(K-2)\rho+1} \sum_{\substack{i=1 \\ i\neq k}}^{K} \hat{n}_t^{(i)} \tag{3.20}$$

$$\begin{aligned}\bar{\sigma}_t^{2\,(k)} &= \sigma^2 \left[\frac{(1-\rho)(1+(K-1)\rho)}{(1+(K-2)\rho)}\right] + \left[\frac{\rho}{(K-2)\rho+1}\right]^2 \times \\ &\left[\sum_{\substack{i=1 \\ i\neq k}}^{K} \left(v_t^{2\,(i)} + \rho^2 \sum_{\substack{j=1 \\ j\neq i}}^{K} v_t^{2\,(j)} \right) + 2 \sum_{\substack{i=1 \\ i\neq k}}^{K-1} \sum_{\substack{j=2 \\ j\neq k \\ j>i}}^{K} E\left\{\bar{n}_t^{(i)} \bar{n}_t^{(j)}\right\} \right. \\ &\left. -2 \sum_{\substack{i=1 \\ i\neq k}}^{K-1} \sum_{\substack{j=2 \\ j\neq k \\ j>i}}^{K} \hat{n}_t^{(i)} \hat{n}_t^{(j)} \right].\end{aligned} \tag{3.21}$$

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 $\mu_t^{(k)} = d + \mu_{t,\,MAI}^{(k)} + \bar{\mu}_t^{(k)}$ and variance $\sigma_t^{2\,(k)} = \sigma_{t,\,MAI}^{2\,(k)} + \bar{\sigma}_t^{2\,(k)}$, where the continuous Gaussian approximation decoder derived in 3.1.1 has been utilized to update probability distribution of the MAI.

[3] The derivations of the following mean and variance can be found in [18].

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 report 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. The simulations are performed using the block sizes of $M = 192$ bits and $M = 600$ bits[1]. All of the simulations are run with $K = 5$ users.

The results for the decoder utilizing continuous approximation of the MAI is 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

[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.

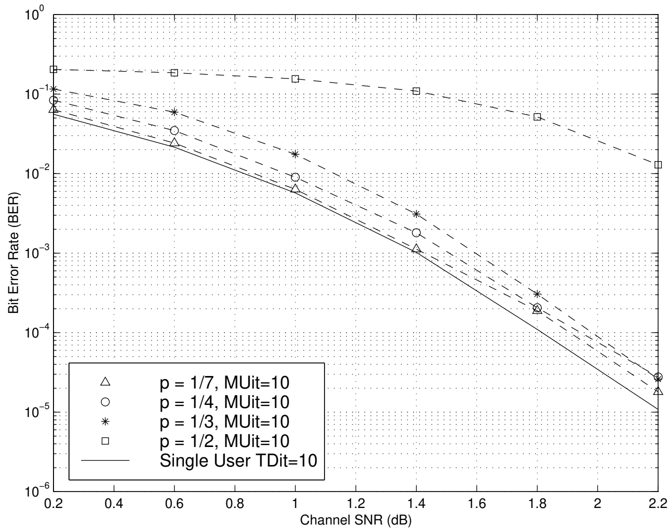

Figure 4.1: Continuous Approximation of MAI with $K = 5$ Users and Block Size $M = 192$ for Various Values of ρ

based on the simplification that the cross-correlation is constant between all users and is time invariant. Figure 4.1 shows the BER performance of the continuous Gaussian approximation of the MAI decoder versus the channel SNR for various values of ρ after the 5^{th} iteration of the multi-user decoder, after 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 ρ increases. Most of 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 3.1.1. 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 5^{th} and 10^{th} 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 10^{th} 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 5^{th} to the 10^{th} 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 same number of iterations.

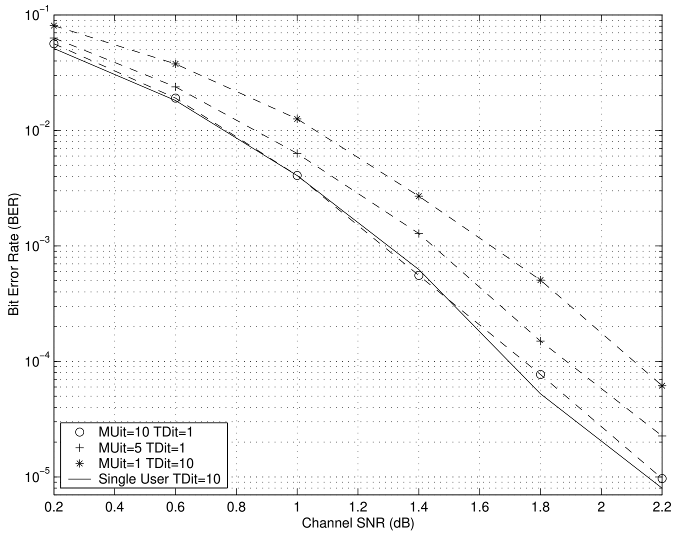

Figure 4.2: Continuous Approximation of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

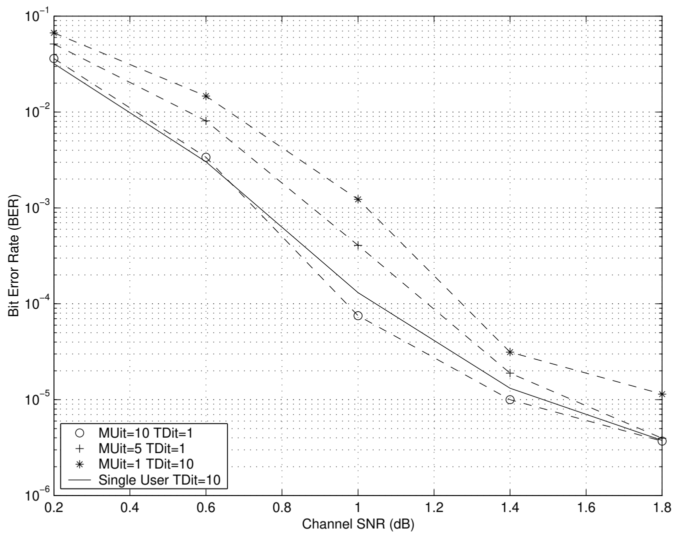

Figure 4.3: Continuous Approximation of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$

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 3.1.2, is shown in Figures 4.4 and 4.5 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^{th} iteration for both block sizes. At a BER of 10^{-3}, the 10^{th} iteration is about 0.15dB better than the 5^{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^{th} to the 10^{th} iteration at larger SNR values for a block size of $M = 600$.

4.4 Continuous Approximation Versus Discrete Analysis of MAI BER Performance

Figures 4.6 and 4.7 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.

4.5 Correlated Gaussian Noise Model BER Performance

The correlated Gaussian noise model derived in section 3.1.3 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 using the

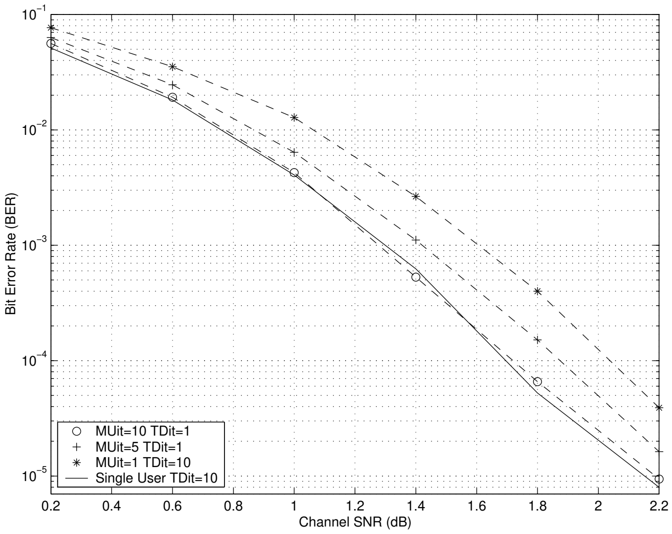

Figure 4.4: Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

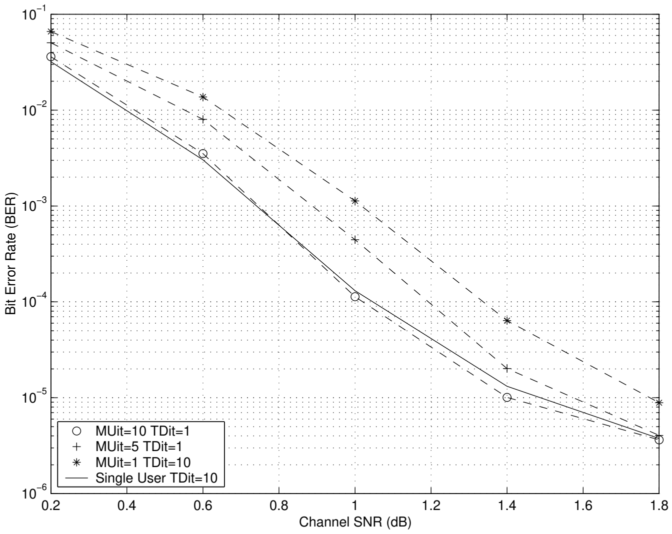

Figure 4.5: Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$

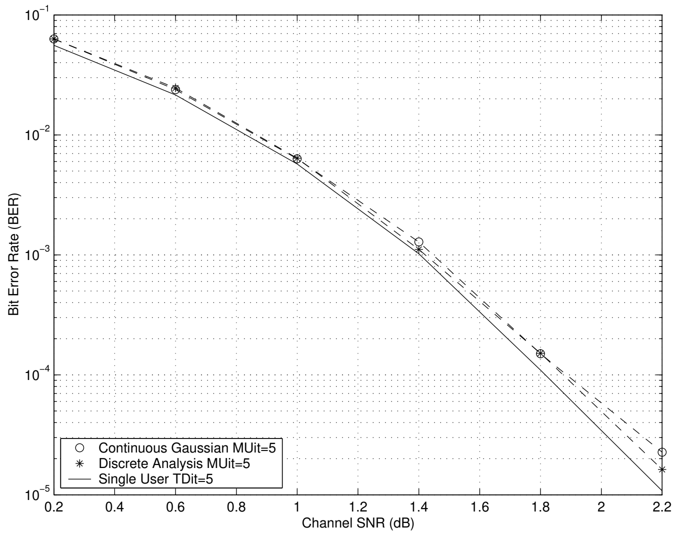

Figure 4.6: Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

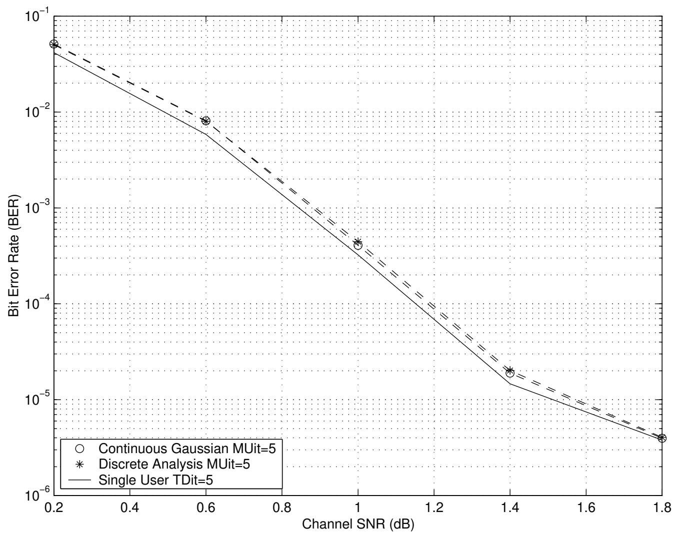

Figure 4.7: Continuous Approximation Versus Discrete Analysis of MAI with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$

continuous Gaussian approximation. The probability distribution of the MAI was chosen as the continuous Gaussian approximation in order to keep the complexity low.

Figures 4.8 and 4.9 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 6^{th} iteration.

The figures show that the updating of the noise probability distribution during the 6^{th} 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 3.1.4 for a block size of $M = 192$ and $M = 600$ is shown in Figures 4.10 and 4.11. Figure 4.10 shows that the extended correlated Gaussian noise decoder performs better than the continuous Gaussian approximation decoder for the 6^{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 4.11, since single-user performance has been achieved at the 6^{th} iteration by both decoders.

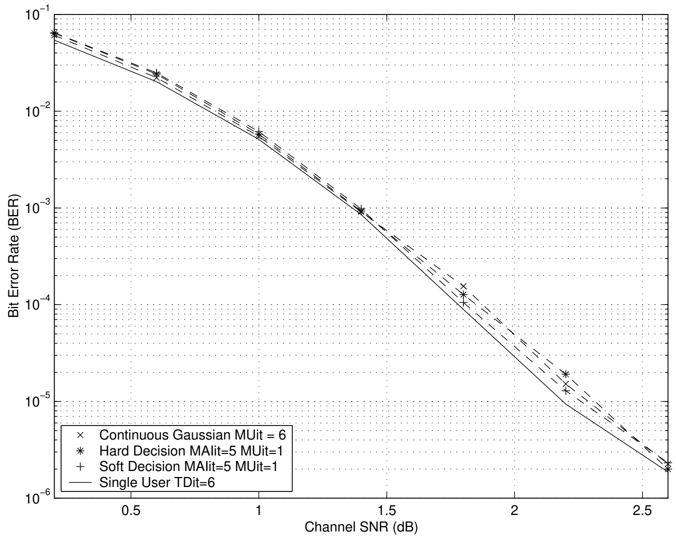

Figure 4.8: Hard Versus Soft Decision of Noise with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

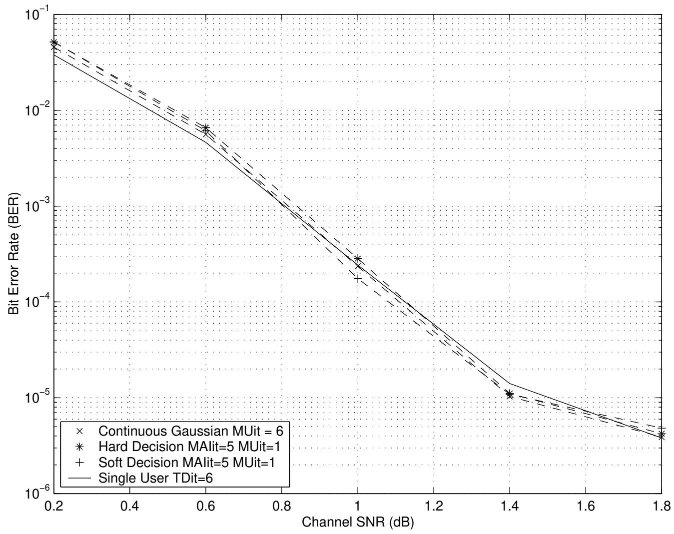

Figure 4.9: Hard Versus Soft Decision of Noise with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$

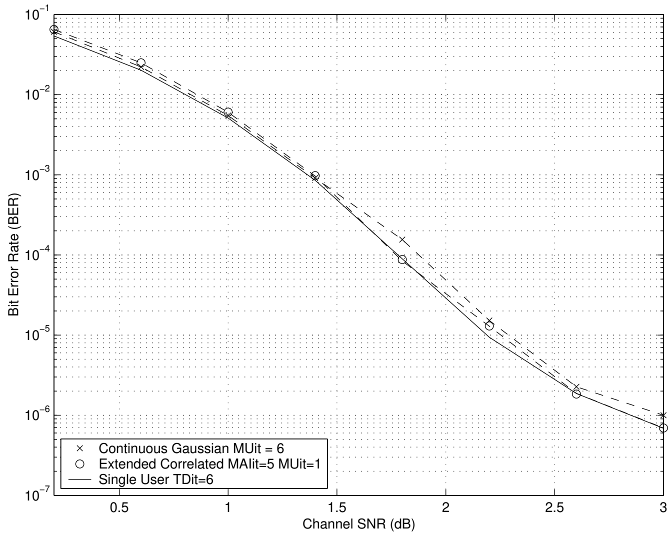

Figure 4.10: Extended Correlated Gaussian Noise Model with $K = 5$ Users, Block Size $M = 192$ and Chip Length $N = 7$

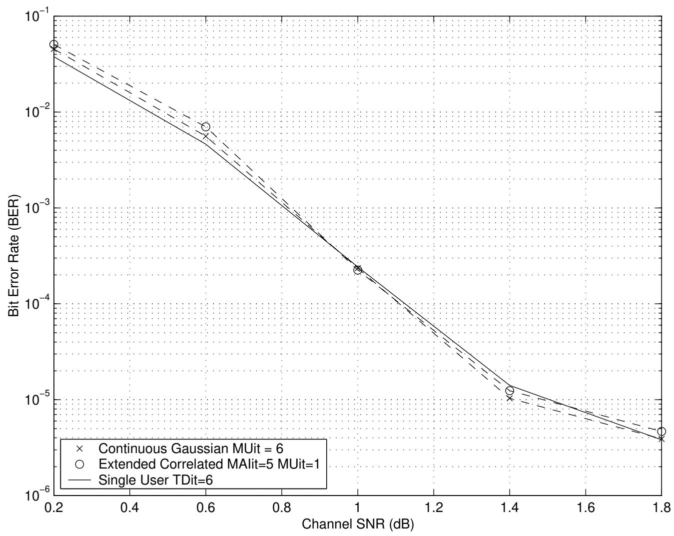

Figure 4.11: Extended Correlated Gaussian Noise Model with $K = 5$ Users, Block Size $M = 600$ and Chip Length $N = 7$

4.7 Complexity Analysis

The computational complexity of the various iterative multi-user CDMA decoders is examined in [18]. 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 sytem. 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 has polymonial complexity proportional to K^3 per multi-user iteration. 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 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 4.6 and 4.7 show the comparison between the continuous Gaussian 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.

Table 4.1 shows that the discrete analysis decoder is 30% more complex than the continuous Gaussian approximation decoder. This complexity difference is expected since the discrete analysis decoder has to perform the computationally intensive convolution operation in order to compute

	$K = 5$ Users
Single-User System	1.00
Continuous Gaussian Approximation of MAI	1.51
Discrete Analysis of MAI	1.81

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

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 [16] 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 continous Gaussian approximation of the MAI decoder along with 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 $= 1$. The performance results for these decoders with $K = 5$ users are shown in Figures 4.8 and 4.9 for the correlated Gaussian noise model at block sizes $M = 192$ and $M = 600$. Figures 4.10 and 4.11 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

	$K = 5$ Users
Single-User System	1.00
Continuous Gaussian Approximation of MAI	1.516
Correlated Gaussian Noise Model	1.523
Extended Correlated Gaussian Noise Model	1.615

Table 4.2: Relative Computational Complexity Per User For Continuous Gaussian Approximation of MAI, Correlated and Extended Correlated Gaussian Noise Model Receivers

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 report 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.

Bibliography

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1)," *Proceedings IEEE International Conference on Communications*, Geneva, Switzerland, pp. 1064-1070, May 1993.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, pp. 284-287, March 1974.

[3] P. Robertson. "Illuminating the Structure of Code and Decoder for Parallel Concatenated Recursive Systematic (Turbo) Codes," *3rd International Conference on Universal Personal Communication*, San Diego, California, pp. 183-187, September 1994.

[4] P. Jung and M. M. Naßhan. "Comprehensive Comparison of Turbo-Code Decoders," *Proceedings of the IEEE Vehicular Technology Conference '95*, Chicago, Illinois, pp. 624-628, June 1995.

[5] J. G. Proakis. "Digital Communications," 3rd Ed., McGraw-Hill, Inc., 1995.

[6] C. E. Shannon. "A Mathematical Theory of Communication," *Bell Syst. Tech. Journal*, vol. 27, pp. 379-423 and 623-656, 1948.

[7] A. J. Viterbi. "CDMA Principles of Spread Spectrum Communication," Addison-Wesley Publishing Company, Inc., 1995.

[8] S. Verdú. "Optimal Multi-User Signal Detection," Ph.D Thesis, University of Illinois at Urbana-Champaign, Coordinated Science Laboratory, Urbana, U.S.A., September 1984.

[9] S. Verdú. "Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, pp. 85-96, January 1986.

[10] S. Verdú. "Computational Complexity of Optimum Multiuser Detection," *Algorithmica*, vol. 4, pp. 303-312, 1989.

[11] S. Verdú. "Multiuser Detection," 1st Ed., Cambridge University Press, 1998.

[12] M. C. Reed, P. D. Alexander, J. A. Asenstorfer, and C. B. Schlegel. "Iterative Multi-User Detection for DS-CDMA with FEC," *The International Symposium on Turbo Codes & Related Topics*, Brest, France, pp. 162-165, September 1997.

[13] M. C. Reed, P. D. Alexander, J. A. Asenstorfer, and C. B. Schlegel. "Near Single User Performance using Iterative Multi-User Detection for CDMA with Turbo-Code Decoders," *IEEE 8^{th} International Symposium on Personal, Indoor, and Mobile Radio Communications*, Helsinki, Finland, pp. 740-744, September 1997.

[14] M. C. Reed, P. D. Alexander, J. A. Asenstorfer, and C. B. Schlegel. "Reduced Complexity Iterative Multi-User Detection for DS/CDMA with FEC," *International Conference on Universal Personal Communications*, San Diego, U.S.A., pp. 10-14, October 1997.

[15] A. Duel-Hallen, J. Holtzman, and Z. Zvonar, "Multiuser Detection for CDMA Systems," *IEEE Personal Communications*, vol. 2, no. 2, pp. 46-58, April 1995.

[16] T. R. Giallorenzi and S. G. Wilson. "Multiuser ML Sequence Estimator for Convolutionally Coded Asynchronous DS-CDMA Systems," *IEEE Transactions on Communications*, vol. 44, pp. 997-1008, August 1996.

[17] P. D. Alexander, A. J. Grant, and M. C. Reed. "Iterative Detection In Code-Division Multiple-Access With Error Control Coding," *European Transactions on Telecommunica-*

tions, Special Issue on CDMA Techniques for Wireless Communications Systems, vol. 9, no. 5, September-October 1998.

[18] D. J. Stienstra. "Iterative Multi-User Turbo-Code Receiver for DS-CDMA," M.A.Sc. Thesis, University of Waterloo, Waterloo, Ontario, January 2000.