Low-Complexity Turbo Multiuser Detection for Coded CDMA

Jagath Jonnalagedda

Problem Report submitted to the Lane Department of Computer Science and Electrical Engineering of the West Virginia University in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Daryl S. Reynolds, Ph.D., Chair
Matthew C. Valenti, Ph.D.
Hong Jian Lai, Ph.D.

2005
Morgantown, West Virginia

Keywords: Turbo Principle, Coded CDMA, Multiuser Detection, Soft interference cancellation

Low-Complexity Turbo Multiuser Detection for Coded CDMA

Jagath Jonnalagedda

ABSTRACT

In this project, a low complexity iterative receiver structure is implemented for decoding the multiuser information in a convolutionally coded synchronous CDMA system in MATLAB. The receiver performs two successive soft output decisions achieved by a soft-input soft output (SISO) multiuser detector and a bank of single user SISO channel decoders, through an iterative process. Such a low complexity SISO multiuser detector is implemented using soft interference cancellation followed by a linear minimum mean square error (MMSE) filtering. SISO channel decoder is developed using log-MAP algorithm. Another low complexity SISO multiuser detector called soft interference cancellation-matched filter (SIC-MF) multiuser detector that skips the linear MMSE filtering stage is also simulated. Simulation results show that at high signal to noise ratio, the performance of both iterative receivers approaches single user performance, but the SIC-MMSE SISO multiuser detector performs better than SIC-MF SISO multiuser detector because it suppresses the residual interference in the channel using the linear MMSE filtering.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee chairman, Dr. Daryl S. Reynolds for his patience, invaluable support and esteemed guidance throughout this project. I would like to express special thanks to my committee member, Dr. Matthew C. Valenti for his valuable suggestions and encouragement. I would also like to thank Dr. Hong Jian Lai for his time.

Table of Contents

List of Figures

Chapter 1

Introduction

In this project, an iterative receiver structure is implemented for decoding multiuser information for coded DS-CDMA system as proposed in [1]. The turbo principle is applied to soft-input soft-output (SISO) multiuser detector and a bank of single user SISO channel decoders before making a bit decision. Applying the turbo principle for decoding gives good performance, approaching the single user performance, while the overall complexity of the system is not increased much. First let us discuss some of the basic concepts in this chapter.

1.1 Channel Capacity

In 1948 Claude Shannon stated a popular theorem called *Noisy Channel Coding Theorem* that, it is possible to achieve an arbitrarily low error probability by transmitting very large blocks of data with transmission rates below the theoretical capacity of the channel. If R > C, where R and C are transmission rate and capacity of the channel respectively, it is not possible to achieve reliable communication with any code [2]. From the above theorem, it can be derived that every channel has certain capacity associated with it and has the major significance because it serves as upper bound on the code rate.

Channel capacity can be defined as the maximum average mutual information between channel's input and output distributions. Shannon (1948b) derived the formula for the capacity of an additive white Gaussian noise (AWGN) channel as:

$$C = W \log_2\left(1 + \frac{P}{W\ N_0}\right) \tag{1.1}$$

Where

C = channel capacity in bits/s

W = transmission bandwidth in Hz

P = average transmitted power in Watts

N_0 = single-sided noise power spectral density in Watts/Hz

From the equation (1.1), it can be seen that the capacity of the channel depends on the transmission bandwidth, signal power and noise power of the channel.

The average transmitted power is given by the expression:

$$P = C\,E_b \tag{1.2}$$

where E_b is energy per bit in Joules.

Hence normalized channel capacity $\frac{C}{W}$ can be expressed as a function of the SNR per bit by replacing the P with $C\,E_b$.

$$\frac{C}{W} = \log_2\left(1 + \frac{C\,E_b}{W\,N_0}\right). \tag{1.3}$$

The expression for $\frac{E_b}{N_0}$ is obtained by solving the equation (1.3)

$$\frac{E_b}{N_0} = \frac{2^{C/W} - 1}{C/W}. \tag{1.4}$$

The plot for the equation (1.4) is shown below in Fig 1.1. From Fig 1.1, at points to the right of the curve, reliable communication is possible and at points to the left of curve, no amount of coding will achieve error free transmission.

Another important point to be noted is, the minimum SNR required to achieve reliable communication when $\frac{C}{W} \to 0$ that is infinite bandwidth condition is given by,

$$\frac{E_b}{N_0} = \lim_{C/W \to 0} \frac{2^{C/W} - 1}{C/W} = \ln 2 = -1.6dB. \tag{1.5}$$

The proof of noisy channel coding theorem assumes random coding and large blocks of data. However, in practical case, decoder for completely random code is impossible, so finite blocks of data are used. In this scenario lower bit error probability can be achieved either by increasing the length of the block or increasing the SNR.

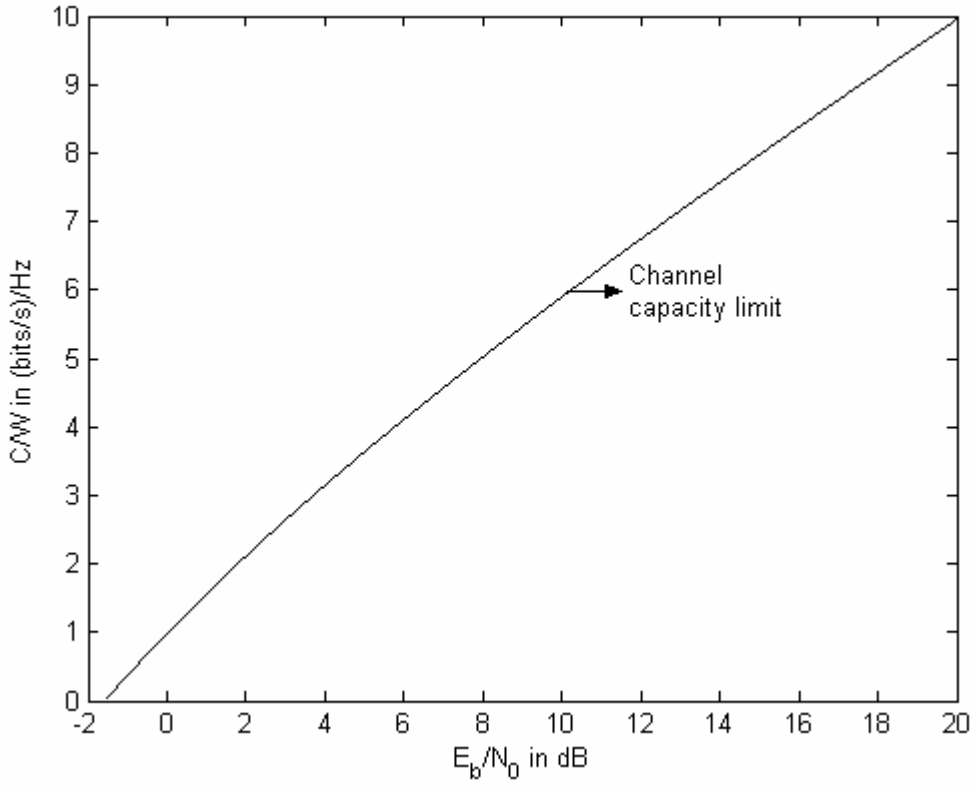

Figure 1.1 Graph for the normalized capacity of the band-limited AWGN channel

1.2 CDMA

In modern communication systems, multiple users want to share the limited radio spectrum simultaneously. There are several multiple access techniques through which mobile users can use the spectrum at same time. *Frequency division multiple access* (FDMA), *time division multiple access* (TDMA) and *code division multiple access* (CDMA) are the major multiple access techniques available. CDMA is based on *spread spectrum* technique.

Spread spectrum is a method of converting the narrowband transmitted signal to a wideband noise-like signal using the sequence called pseudo noise (PN) sequence [3]. This PN sequence should be independent of the information data. The signal used to spread the bandwidth of the transmitted signal is called the spreading signal. The amount of the signal spread is called the *processing gain* denoted by "N". The two main types of spread spectrum techniques are *direct sequence spread spectrum* (DS/SS) and *frequency hopping spread spectrum* (FH/SS). In this report, discussion is restricted to DS/SS.

In direct sequence spread spectrum, the data signal is multiplied by the wideband PN signal to spread the signal. Direct sequence spread spectrum is mostly used in CDMA that allows multiple mobile users to share the spectrum at same time and frequency as shown in Fig 1.2.

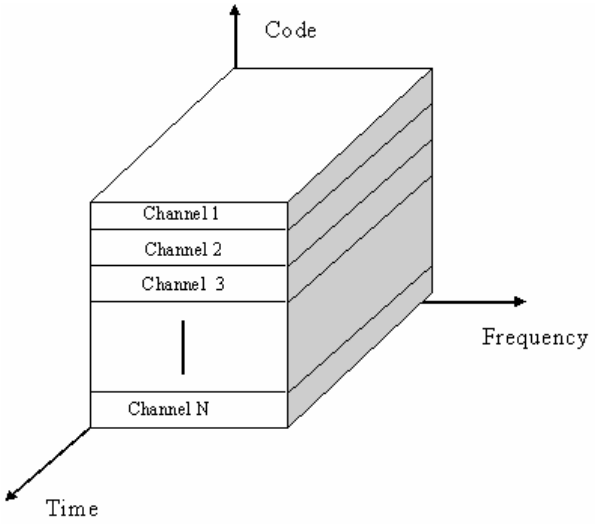

Figure 1.2 Channel usage by CDMA [3]

In DS-CDMA system, each user is identified by its own code also called as *signature sequence* of N chips. In order to prevent users from interacting with each other, these codes are designed to be orthogonal to each other. But in reality perfect orthogonal codes are relatively few within a given bandwidth. Multipath fading in the channel destroys the orthogonality and hence cross correlation between two codewords is not exactly zero. To detect the user, the receiver is required to know the signature sequence used by the transmitter and performs time correlation to detect only the signature sequence of the desired user and assumes all other codewords as noise. A simple DS-CDMA system is shown in Fig 1.3.

The PN code is a sequence of chips values -1 or 0 and 1. The number of chips within one code is called the processing gain of the code. The PN sequences are generated using linear feedback shift registers (LFSR) characterized by either the Fibonacci (simple) or Galois (modular) series. In both the cases, shift register generator is a finite state machine represented by an irreducible primitive polynomial of the form: [4]

$$f(x) = x^n + c_{n-1}\, x^{n-1} + ...+ c_2\, x^2 + c_1\, x + 1. \tag{1.6}$$

In the equation (1.6), $\{c_i\}$ are set of non-zero coefficients, where $c_i = 1$ represents a connection and $c_i = 0$ represents no connection in LFSR. If such register has n-flip-flops, then it generates a

set of $L = 2^{n-1}$ periodic sequences of length $L = 2^{n-1}$. These L sequences are shifted version of the initial sequence of length L.

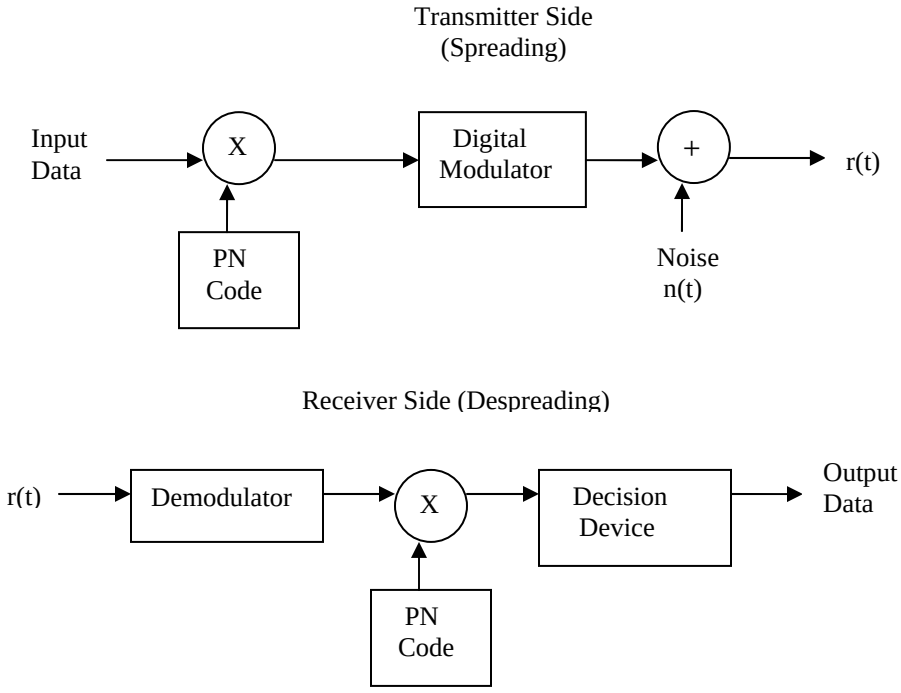

Figure 1.3 Transmitter and Receiver Structure for DS-CDMA system

The PN sequence generated by LFSR with the period 2^{n-1} is called a *maximal length sequence* or *m-sequence*. Another type of PN sequences are *Gold codes* which are generated by the modulo-2 addition of two m-sequences of the same length. Perfect orthogonal PN sequences are generated by *Hadamard-Walsh codes*. These are set of $N = 2^n$ codes with length $N = 2^n$:

$$H_N = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \quad \text{where } H_0 = [1]. \tag{1.7}$$

Advantages of CDMA are:

- Can always increase the number users using the system, but also increases noise, "*graceful degradation*".

- Gives good performance in multipath fading as the signal is spread over a wide spectrum because it allows for exploitation of multipath diversity via RAKE reception.
- Spread spectrum CDMA is also resistant to narrowband interference. The despreading of the signal at the receiver spreads of the narrowband signal over a wide bandwidth. So, after despreading, the narrowband interferer becomes wideband and the original data signal becomes narrowband signal. A bandpass filter can be used to allow only the narrowband signal which is the desired signal so that only the interferer power that falls in the bandwidth of the despread signal causes any interference.
- More secure in the sense that, without the knowledge of the spreading code, it is difficult to recover the data, "*low probability of intercept*".

Disadvantages of CDMA are:

- Suffers from *near-far* problem. It appears when the closest transmitter is able to capture the receiver because of the small propagation path loss and it can be reduced using proper power control.
- Problem of *self jamming* caused by non-orthogonal PN sequences. This problem can be overcome by multiuser detection.

1.3 Overview of the Report

Multiple access interference (MAI) cause major problems in CDMA channels for reliable communication. In this report an iterative receiver structure is simulated based on turbo multiuser detection as proposed in [1]. Turbo multiuser detection is a joint channel decoding and multiuser detection using an iterative exchange of soft information between the two. In this a low complexity SISO multiuser detector is implemented which makes use of both soft interference cancellation and instantaneous linear minimum mean square filtering given in [1]. Also this performance is compared with the results obtained when MMSE SISO detector is replaced with a SISO matched filter given in [5].

Rest of the report is organized as follows: Chapter 2 describes the channel coding. Encoding and soft decision decoding of convolutional codes is explained. Chapter 3 gives the concepts of matched filter and MMSE multiuser detector. Chapter 4 explains turbo principle, turbo multiuser detection. Chapter 5 presents the results obtained and chapter 6 contains conclusions and possible future work.

Chapter 2

Channel Coding

As discussed in chapter 1, Shannon proved that reliable communication over a noisy channel is possible through random coding. Since then many coding schemes have been developed. All these codes have some structure making possible to implement them. Block codes, convolutional codes and turbo codes are some examples of practical channel codes. This chapter concentrates only on convolutional codes.

Channel codes add some redundancy to the information bits in a controlled manner that can be used at the receiver to detect or correct the errors caused by the noise in the channel. The method of knowing which bits are incorrect is called *forward error correction*. If k represents the number of input data bits, then encoder maps each k-bit sequence into a unique n-bit sequence. The *code rate* of such (n, k) code is given by k/n. Before going into the further details of the convolutional encoding, let us define some parameters of the channel coding process.

If the input-bit sequences are represented by $\mathbf{b}$ and output sequences by $\mathbf{C}$ and if the code is linear, then

$$\mathbf{C} = \mathbf{b}\mathbf{G} \tag{2.1}$$

where $\mathbf{G}$ is called g*enerator matrix* of size $k \times n$ and the multiplication is modulo-2 multiplication. Some important definitions are:

- *Hamming distance* between any two code words is defined as the number of bit positions that two code words differ.
- The *minimum distance* $d_{\min}$ is the minimum Hamming distance between any two distinct code words.
- *Hamming weight* is the number of 1's in the code word.
- A code is said to *linear* if the modulo-2 addition of any two code words is also a code word.
- A *systematic code* is a code in which a part of the code word is data bits.
- A code is *cyclic* if cyclic shift of a code word is also a valid code word.
- The *error correcting capability* is defined as the minimum number of errors in a code word that a code can correct and is given my the expression: [6]

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \tag{2.2}$$

Convolutional codes operate on serial data, one or few bits at a time. In block codes, output is independent of the previous data blocks, where as in convolutional codes; the output code word is a linear combination of information bits in the current block and a selected number of preceding blocks.

2.1 Convolutional Encoding

Convolutional encoder continuously encodes a stream of information bits. Shorthand notation of a convolutional code is (n, k, l), where n is number of streams at output of the encoder, k is number of streams at the input to the encoder and l is called *constraint length* which is equal to the sum of the number of input streams and the number of memory elements in the encoder.

A convolutional encoder is represented in different ways, with a generator polynomial, encoder block diagram, state and trellis diagrams. The convolutional code used in this project is (2, 1, 5) code that is rate ½ convolutional code with constraint length of 5. Generator polynomials for this code are given below. Octal representation of these polynomials is (23, 35).

$$g_1 = 1 + D^3 + D^4 \tag{2.3}$$

$$g_2 = 1 + D + D^2 + D^4 . \tag{2.4}$$

Convolutional encoder block diagram is shown in Fig 2.1.

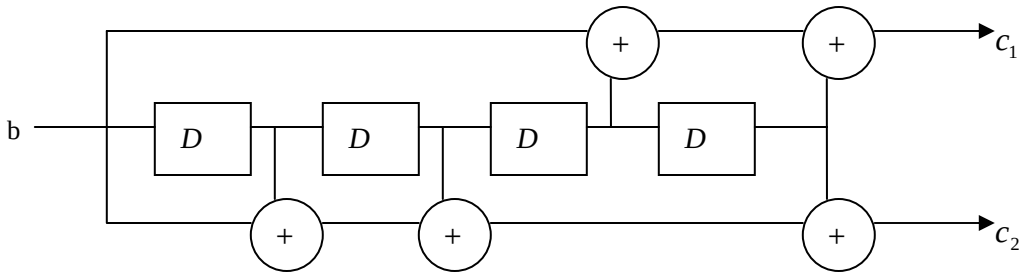

Figure 2.1 Rate ½, (23, 35) convolutional encoder with constraint length of 5.

As shown in Fig 2.1, a convolutional encoder is implemented via linear shift registers with memory elements represented by the letter '*m*' and the modulo-2 adders. Initially all the memory elements are flushed with zeros. These memory elements are also called as delay elements. Which of the delay elements are connected to the modulo-2 adders is determined by the generator

polynomial given in equations (2.3) and (2.4). When an input bit stream is inserted at the input of the encoder block diagram, the first information bit is pushed to the first delay element D^1. Every time a new information bit is inserted into this shift register, the previous bit that is already present in that register is shifted to the right for one bit position. The shifted bits are added (modulo-2 addition) or XOR'ed through respective connections and produces output codebit bits. Since it is rate ½ convolutional encoder, each information bit inserted into the block diagram produces two bits of codeword. It is a continuous process till the end of the input stream.

The state diagram is used to show the relationship between the encoder state, input and output. The state diagram will have 2^m nodes, where m is the number of memory elements in the convolutional encoder and these nodes are connected by branches. Every node will have two branches coming in and two branches going out.

Though state diagram shows all the possible state transitions of the convolutional encoder, it does not contain state transitions over particular time. The trellis diagram clearly shows this. There will be 2^m number of states in the trellis diagram, where m is the number of memory elements in the convolutional encoder. For a simple example, consider a (2, 1, 3) convolutional code with generators (5,7) in octal. Hence there will be two memory elements and the number of states in encoder will be four. Trellis diagram for this code is shown in Fig 2.2.

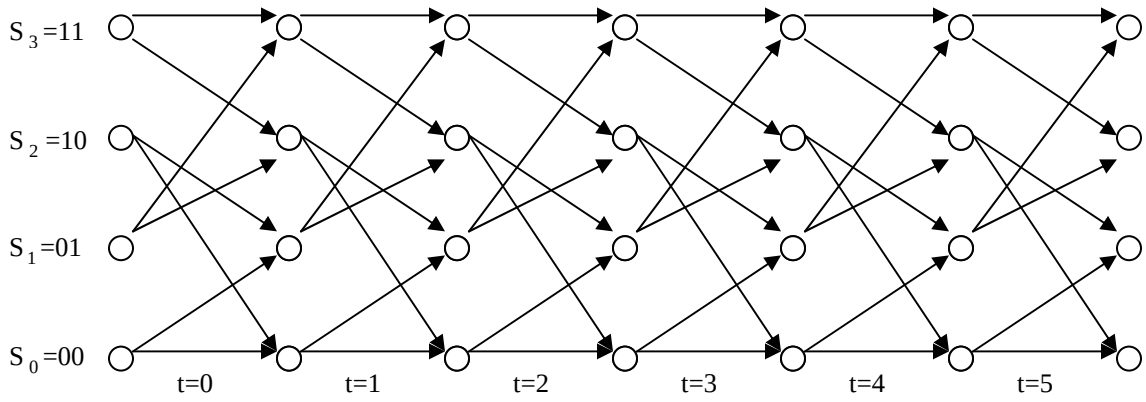

Figure 2.2 Trellis Diagram for (2, 1, 3) convolutional code with generators (5, 7) in octal

Here the trellis length of six is considered. All 4 possible states for (2, 1, 3) convolutional code are shown at each instant of time. A movement to the right through the trellis indicates the passage of time. Each state has two branches coming in and two branches are going out. So the number of jumps from one state to another state is limited and hence information bit stream

encoded with convolutional encoder must follow a certain path and this path can be found using the trellis diagram.

2.2 Convolutional Decoding

When this codebit stream is transmitted through a channel, the receiver has to decode a sequence that can be corrupted by the noise and fading in the channel. The decoding process can be broadly classified as hard decision decoding and soft decision decoding. In hard decision decoding, hard decision is made on channel output and the goal of this type of decoding is to correct binary errors introduced in the hard decision process. Where as in soft decision decoding, the channel outputs are directly sent to the decoder and thus soft decision decoding is more complex than hard decision decoding, but it gives much better performance than the earlier one.

The most popular hard decision decoding algorithm is *Viterbi algorithm* [7]. The Viterbi algorithm (VA) tries to find the most likely path through the trellis and hence called Maximum Likelihood (ML) decoding. Another alternative decoding algorithm to VA is symbol-by-symbol *maximum a posteriori* (MAP) algorithm, which is given in [8], also called as BCJR algorithm. The main difference between these two algorithms is, VA tries to find the most probable state sequence and the MAP algorithm determines the most likely individual state transition given the noisy channel output. As discussed earlier VA does not produce soft outputs, so the modification to the Viterbi algorithm is suggested in [9] to give the soft outputs in terms of a posteriori bit probabilities. This algorithm is called *soft output Viterbi algorithm* (SOVA). Finally the soft decision algorithms derived based on MAP algorithm are *log-MAP* algorithm and *max-log-MAP* algorithm [10]. These two algorithms operate in log domain. In this project log-MAP algorithm is used for channel decoding and is described in next section. An excellent reference for all these algorithms can be found in [6]. The following description of the log-MAP algorithm is based on this reference [6] only.

2.2.1 Log-MAP Algorithm

Log-MAP algorithm can be termed as *soft input soft output* (SISO) decoding algorithm. A decoding algorithm that accepts soft inputs from the demodulator called a-priori information and produces soft outputs called a-posteriori information at its output is called as soft input, soft output decoding algorithm. The reliability of a decoded bit is best represented by the a-posteriori

probability (APP), $P(d/\mathbf{y})$. Soft output that is an estimate of information bit d (+1 / 0) having the received vector $\mathbf{y}$ is given by,

$$\Lambda_i = \ln\left(\frac{P[d_i = 1|\,\mathbf{y}]}{P[d_i = 0|\,\mathbf{y}]}\right) \tag{2.5}$$

where, Λ_i is called the *log-likelihood ratio* (LLR). As discussed earlier, log-MAP algorithm operates in log domain, but for now, all the following components are defined with respect to MAP algorithm. Later we apply natural logarithm to these components.

First of all, the MAP algorithm finds the probability $P[s_i \to s_{i+1}|\,\mathbf{y}]$, where s_i represents the state of the trellis at time i, of each valid state transmission in the trellis diagram given the received vector $\mathbf{y}$. From the definition of conditional probability,

$$P[s_i \to s_{i+1}|\,\mathbf{y}] = \frac{P[s_i \to s_{i+1},\,\mathbf{y}]}{P[\mathbf{y}]}. \tag{2.6}$$

Now using the Markov process, the numerator in the equation (2.6) can be divided into [6]

$$P[s_i \to s_{i+1},\,\mathbf{y}] = \alpha(s_i)\,\gamma(s_i \to s_{i+1})\,\beta(s_{i+1}) \tag{2.7}$$

where, $$\alpha(s_i) = P[s_i,\ (y_0, y_1,\ ..., y_{i-1})]. \tag{2.8}$$

The term $\alpha(s_i)$ represents the probability that the current state is s_i given the channel output $y_0, y_1, ..., y_{i-1}$. The final state s_{i+1} is not considered here since the current state does not depend on the future values of the channel output. The second term on right hand side of equation (2.7) is

$$\gamma(s_i \to s_{i+1}) = P[s_{i+1}, y_i|\,s_i]. \tag{2.9}$$

Equation (2.9) gives the probability that the state transition form state s_i to state s_{i+1}, given the current state is s_i. The term $\gamma(s_i \to s_{i+1})$ is called as the branch metric associated with this state transition. The branch metric is calculated as follows:

$$\begin{aligned}\gamma(s_i \to s_{i+1}) &= P[s_{i+1}|\,s_i]\,P[y_i|s_i \to s_{i+1}] \\ &= P[d_i]P[y_i|\,c_i]\end{aligned} \tag{2.10}$$

where d_i and c_i are the information and coded bits respectively associated with the sate transition $s_i \rightarrow s_{i+1}$. The branch metric is zero if the states s_i and s_{i+1} are not connected in the trellis diagram. The first term on the right hand side of the equation (2.10) is calculated as,

$$\begin{aligned} P[d_i] &= \frac{e^{z_i}}{1+e^{z_i}}, \; d_i = 1 \\ &= \frac{1}{1+e^{z_i}}, \; d_i = 0 \end{aligned} \tag{2.11}$$

z_i is the a-priori information to the decoder in LLR form. The second term on the right hand side of the equation (2.10) is obtained in Gaussian noise as,

$$P[y_i \mid c_i] = \frac{1}{\sqrt{\pi N_0 E_s}} \exp\left\{ \frac{-E_s}{N_0} \sum_{q=0}^{n-1} [y_i^q - a_i^q (2c_i^q - 1)]^2 \right\} \tag{2.12}$$

here E_s is the energy of the code symbol, a is the fading amplitude and is constant for AWGN noise. So the branch metric is nothing but the Euclidean distance between the received vector and the coded symbol. Where as Hamming distance is used to calculate branch metric in hard decision decoding.

Finally, $$\beta(s_{i+1}) = P[(y_{i+1}, ..., y_{L-1}) | s_{i+1}] \tag{2.13}$$

where *L* is the length of the trellis. This represents the probability that the final state is s_{i+1}.

The $\alpha(s_i)$ is found using the forward recursion

$$\alpha(s_i) = \sum_{s_{i-1} \in \mathrm{A}} \alpha(s_{i-1}) \, \gamma(s_{i-1} \rightarrow s_i) \tag{2.14}$$

where the "A" is the set of states s_{i-1} that are connected to the state s_i as shown in Fig 2.3.

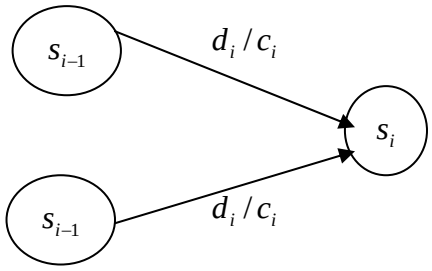

Figure 2.3 Forward recursion

Similarly, $\beta(s_{i+1})$ is found by backward recursion,

$$\beta(s_i) = \sum_{s_{i+1} \in \mathrm{B}} \beta(s_{i+1}) \, \gamma(s_i \rightarrow s_{i+1}) \tag{2.15}$$

where the "B" is the set of states s_{i+1} that are connected to the state s_i as depicted in Fig 2.4.

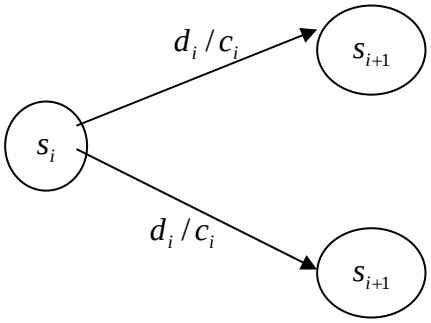

Figure 2.4 Backward recursion

Now the information bit probabilities at each time instant are found using

$$P[d_i = 1|\mathbf{y}] = \sum_{S_1} P[s_i \to s_{i+1}|\mathbf{y}] \tag{2.16}$$

and

$$P[d_i = 0|\mathbf{y}] = \sum_{S_0} P[s_i \to s_{i+1}|\mathbf{y}] \tag{2.17}$$

where S_1 is the set of all state transitions for which the input information bit is "1" and S_0 is the set of all state transitions for which the input information bit is "0". So the LLR of the information bit is given by

$$\Lambda_i = \ln \frac{\sum_{S_1} \alpha(s_i)\, \gamma(s_i \to s_{i+1})\, \beta(s_{i+1})}{\sum_{S_0} \alpha(s_i)\, \gamma(s_i \to s_{i+1})\, \beta(s_{i+1})}. \tag{2.18}$$

One final step before taking all these components into log-domain is defining the *Jacobian logarithm*:

$$\begin{aligned} \ln(e^a + e^b) &= \max(a, b) + \ln(1 + \exp\{-|b - a|\}) \\ &= \max^*(a, b). \end{aligned} \tag{2.19}$$

The benefit of applying logarithm to the multiplication is that multiplication becomes addition. So the "max* "operation is the way of performing addition in log-domain.

Let $\bar{\gamma}(s_i \to s_{i+1})$ be the natural logarithm of $\gamma(s_i \to s_{i+1})$, that is

$$\begin{aligned} \bar{\gamma}(s_i \to s_{i+1}) &= \ln(\ \gamma(s_i \to s_{i+1})) \\ &= \ln(P[d_i]) + \ln(P[y_i|c_i]). \end{aligned} \tag{2.20}$$

$\overline{\alpha}(s_i)$ be the natural logarithm of $\alpha(s_i)$, then

$$\overline{\alpha}(s_i) = \ln(\ \alpha(s_i)\ )$$

$$= \ln\left\{ \sum_{s_{i-1}\in \mathrm{A}} \exp[\,\overline{\alpha}(s_{i-1}) + \overline{\gamma}(s_{i-1}\to s_i)\,] \right\}$$

$$= \max_{s_{i-1}\in \mathrm{A}}{}^{*}[\ \overline{\alpha}(s_{i-1}) + \overline{\gamma}(s_{i-1}\to s_i)\ ] \tag{2.21}$$

and $\overline{\beta}(s_i)$ be the natural logarithm of $\beta(s_i)$, then

$$\overline{\beta}(s_i) = \ln(\ \beta(s_i)\ )$$

$$= \ln\left\{ \sum_{s_{i+1}\in \mathrm{B}} \exp[\,\overline{\beta}(s_{i+1}) + \overline{\gamma}(s_i\to s_{i+1})\,] \right\}$$

$$= \max_{s_{i+1}\in \mathrm{B}}{}^{*}[\ \overline{\beta}(s_{i+1}) + \overline{\gamma}(s_i\to s_{i+1})\ ]. \tag{2.22}$$

The LLR of the information bit can be found using,

$$\Lambda_i = \ln\left\{ \sum_{S_1} \exp[\ \overline{\alpha}(s_i) + \overline{\gamma}(s_i\to s_{i+1}) + \overline{\beta}(s_{i+1})]\right\} - \ln\left\{ \sum_{S_0} \exp[\ \overline{\alpha}(s_i) + \overline{\gamma}(s_i\to s_{i+1}) + \overline{\beta}(s_{i+1})]\right\} \tag{2.23}$$

$$= \max_{S_1}{}^{*}[\,\overline{\alpha}(s_i) + \overline{\gamma}(s_i\to s_{i+1}) + \overline{\beta}(s_{i+1})] - \max_{S_0}{}^{*}[\,\overline{\alpha}(s_i) + \overline{\gamma}(s_i\to s_{i+1}) + \overline{\beta}(s_{i+1})]. \tag{2.24}$$

Implementation of the log-MAP algorithm consists of 3 main steps

Step 1: Forward Recursion

(a) Initialization of $\bar{\alpha}(j,i)$, where $0 \le j \le 2^m - 1, 0 \le i \le L$ as, L is the length of the trellis

$$\bar{\alpha}(j,0) = 0 \text{ if } j = 0$$
$$= -\infty \text{ if } j \neq 0 \tag{2.25}$$

(b) For $i = 1$ *to* L

(c) For $j = 0$ *to* 2^m

(d) Let $s_i = S_j$

(e) Update $\bar{\alpha}(j,i)$ using the equation (2.21), that is

$$\bar{\alpha}(j,i) = \max_{s_{i-1} = S_{j'} \in \mathrm{A}} * [\, \bar{\alpha}(j', i-1) + \bar{\gamma}(s_{i-1} \to s_i) \,] \tag{2.26}$$

here the "A" is the set of states s_{i-1} that are connected to the state s_i.

(f) $j = j + 1$, if $j = 2^m$ then go to step 1 (g), else go to step 1 (d)

(g) $i = i + 1$, if $i = L$, then go to step 2, else go to step 1 (c)

Step 2: Backward Recursion

(a) Initialization of $\bar{\beta}(j,i)$, where $0 \le j \le 2^m - 1, 0 \le i \le L$ as, L is the length of the trellis

$$\bar{\beta}(j,L) = 0 \text{ if } j = 0$$
$$= -\infty \text{ if } j \neq 0 \tag{2.27}$$

(b) For $i = L - 1$ *to* 0

(c) For $j = 0$ *to* 2^m

(d) Let $s_i = S_j$

(e) Update $\bar{\beta}(j,i)$ using the equation (2.22), that is

$$\bar{\beta}(j,i) = \max_{s_{i+1} = S_{j'} \in \mathrm{B}} * [\, \bar{\beta}(j', i+1) + \bar{\gamma}(s_{i-1} \to s_i) \,] \tag{2.28}$$

here the "B" is the set of states s_{i+1} that are connected to the state s_i.

(f) $j = j + 1$, if $j = 2^m$ then go to step 2 (g), else go to step 2 (d)

(g) $i = i - 1$, if $i = 0$, then go to step 3, else go to step 2 (c)

Step 3: Calculation of LLR of the information bits for all $i = 0, 1, ..., L-1$

$$\Lambda_i = \max_{S_1}{}^{*}[\bar{\alpha}(j, i) + \bar{\gamma}(s_i \to s_{i+1}) + \bar{\beta}(j', i+1)] - \max_{S_0}{}^{*}[\bar{\alpha}(j, i) + \bar{\gamma}(s_i \to s_{i+1}) + \bar{\beta}(j', i+1)] \quad (2.29)$$

where S_1 is the set of all state transitions for which the input information bit is "1" and S_0 is the set of all state transitions for which the input information bit is "0".

Note that in *max-log-MAP algorithm*, $\max^{*}(a, b) = \max(a, b)$. The correction factor, $f_c = \ln(1+\exp\{-|b-a|\})$ is omitted in max-log MAP algorithm. It is reasonable because, the correction factor approaches zero when a and b becomes dissimilar. Therefore

$$\ln(e^a + e^b) \approx \max(a, b). \quad (2.30)$$

The complexity of max-log-MAP and log-MAP algorithms is similar. But when compared to Viterbi algorithm, these algorithms are slightly more than twice as complex.

2.3 Performance Comparison

In this section performance of (2,1,5) convolutional code is compared using both hard decision decoding and soft decision decoding. Fig 2.5 shows simulated BER plots for (23, 35) convolutional code with constraint length 5 using Viterbi algorithm and the log-MAP algorithm for BPSK modulation in AWGN noise. It also shows the BER plot for uncoded BPSK. Approximately, hard decision coding gain at BER = 10^{-5} is 2.3 dB and soft decision coding gain at BER = 10^{-5} is 4.5 dB and also it can be seen that soft decision decoding improves the performance of convolutional coding by about 2.2 dB.

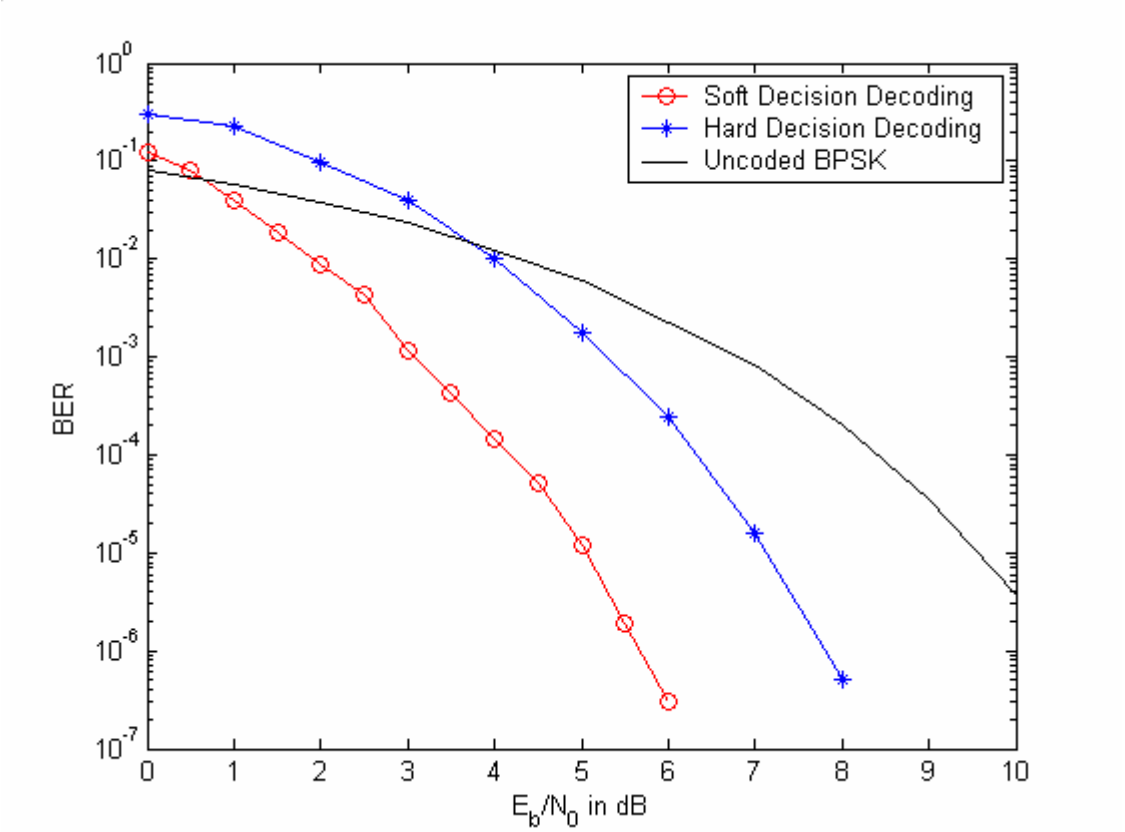

Figure 2.5 Performance comparison of (2, 1, 5) convolutional code in AWGN noise

Fig. 2.6 shows the analytical BER plots of for (23, 35) convolutional code with constraint length 5 with hard decision decoding and soft decision decoding using BPSK modulation.

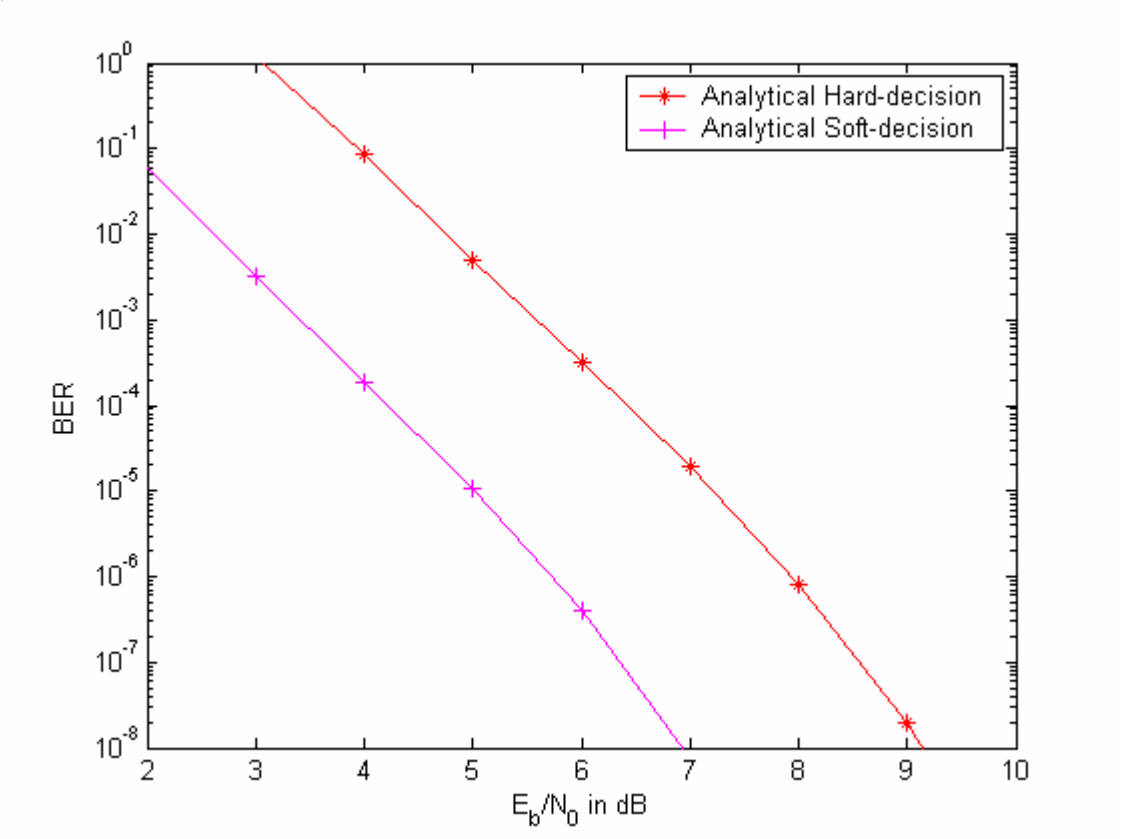

Figure 2.6 Performance comparison of (2, 1, 5) convolutional code with hard decision decoding and soft decision decoding

Chapter 3

Multiuser Detection

Multiuser detection is a broad field that deals with the detection of multiple users whose received signals are not orthogonal to one another. Systems using CDMA have large number of users whose signals overlap in time and frequency. Also system supporting multiple users simultaneously has the inherent problem of multiple access interference (MAI) and a system like DS-CDMA also suffers from near-far effect.

But all these interfering signals contain as much information and structure as the desired signal of interest and hence need not be treated as noise. Multiuser detection (MUD) intelligently exploits this structure and aims at canceling the negative effect a user's signal has on the rest, thus increasing system capacity. Multiuser detection can be implemented as joint demodulation and interference cancellation for improved detection and estimation of a desired signal. MUD achieves better performance than conventional detector. The concept of conventional detector is discussed in next section.

3.1 Matched Filter

Matched filtering with respect to the spreading code is the simplest way of demodulating CDMA signals. It is often called as conventional detector. Conventional demodulators employ a filter matched to the particular user's signature sequence. In case of multiuser system, the demodulator consists of bank of matched filters each matched to the signature waveform of different users in case of CDMA as shown in Fig 3.1.

Consider the basic synchronous CDMA K-user channel model [11],

$$r(t) = \sum_{k=1}^{K} A_k\, b_k\, s_k(t) + n(t), \quad t \in [0, T] \tag{3.1}$$

where,

- $r(t)$ is the received signal
- $A_k > 0$ is the received amplitude of the k^{th} user's signal
- $b_k \in \{+1, -1\}$ is the input bit vector corresponding to the k^{th} user
- $s_k(t)$ is the unit energy signature waveform of the k^{th} user

- $n(t) \sim N(0, \sigma^2 I_N)$ is the Additive White Gaussian Noise
- σ^2 is the variance of the ambient channel noise
- T is the inverse of the data rate.

The spreading waveform $s_k(t)$ is:

$$s_k(t) = \sum_{p=1}^{N} u_k^p \psi(t - pT_c) \tag{3.2}$$

where $(u_k^1, ..., u_k^N)$ is the spreading sequence of length $N = T/T_c$, $\psi(t)$ is the pulse-shaping filter and N is the processing gain. For convenience, assume

$$\psi(t) = 1, \ 0 < t < T_c$$
$$= 0, \ \text{otherwise.} \tag{3.3}$$

The spreading waveforms are normalized. Therefore,

$$\int_0^T s_k^2(t)\, dt = \int_0^T \left(\sum_{p=1}^{N} u_k^p \psi(t - pT_c) \right)^2 dt$$
$$= \sum_{p=1}^{N} (u_k^p)^2$$
$$= 1, \tag{3.4}$$

which implies that $u_k^p \in \left\{ -\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}} \right\}$.

Discrete time synchronous model for equation (3.1) is given by,

$$r[i] = S A b[i] + n[i], \quad i = 1, 2, ..., M \tag{3.5}$$

- $r = [r_1(i), ..., r_K(i)]^T$
- $S = [s_1, ..., s_K]$
- $A = diag\{A_1, ..., A_K\}$
- $b[i] = [b_1(i), ..., b_K(i)]^T$
- M is the number of symbols per user.

The spreading waveform s_k in vector form is given by,

$$s_k = \frac{1}{\sqrt{N}} [u_k^1 .... u_k^N]^T, \ u_k^p \in \{1, -1\}. \tag{3.6}$$

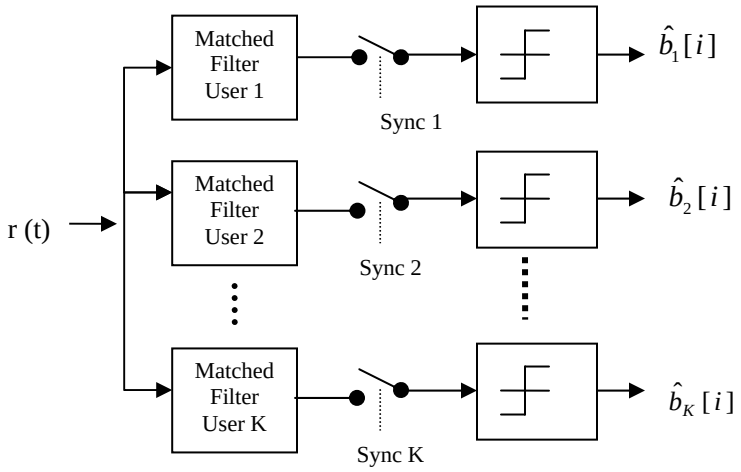

Figure 3.1 Bank of single user matched filters [11]

In matched filtering exact knowledge of the signature sequences is required to implement the detector. So the detector in this case is given by

$$w_k = s_k \tag{3.7}$$

and hence the estimate of the k th bit is,

$$\hat{b}_k = sign\,(w_k^T \,.\, r[i])\,. \tag{3.8}$$

The main drawback of conventional detector is that it neglects MAI as white noise. Hence the performance of matched filter becomes poor for more of number users since as the number of users increases, MAI increases. Another serious limitation of matched filter is it is seriously affected by the near-far problem.

This detection strategy is optimal in single user transmission in an AWGN channel and in multi-user orthogonal and synchronous communication. In a practical multiple access channel scenario, this is not possible. Hence there is a need for MUD. The multiuser detector exploits the cross-correlations between the signals, uses it to reduce MAI, and thereby improves data detection.

3.2 Classification of Multiuser Detectors

Multiuser detectors can be *optimal or suboptimal*. Classification of multiuser detectors is shown in Fig 3.2. The optimal multiuser detectors consist of a bank of matched filters at the front end

followed by a dynamic programming algorithm [11]. This decision algorithm can be either Viterbi algorithm or BCJR algorithm. The optimal multiuser detector is shown in Fig 3.3. Though they give optimal theoretical performance, complexity increases exponentially with the number of users and hence are not suitable for CDMA systems with a large number of users.

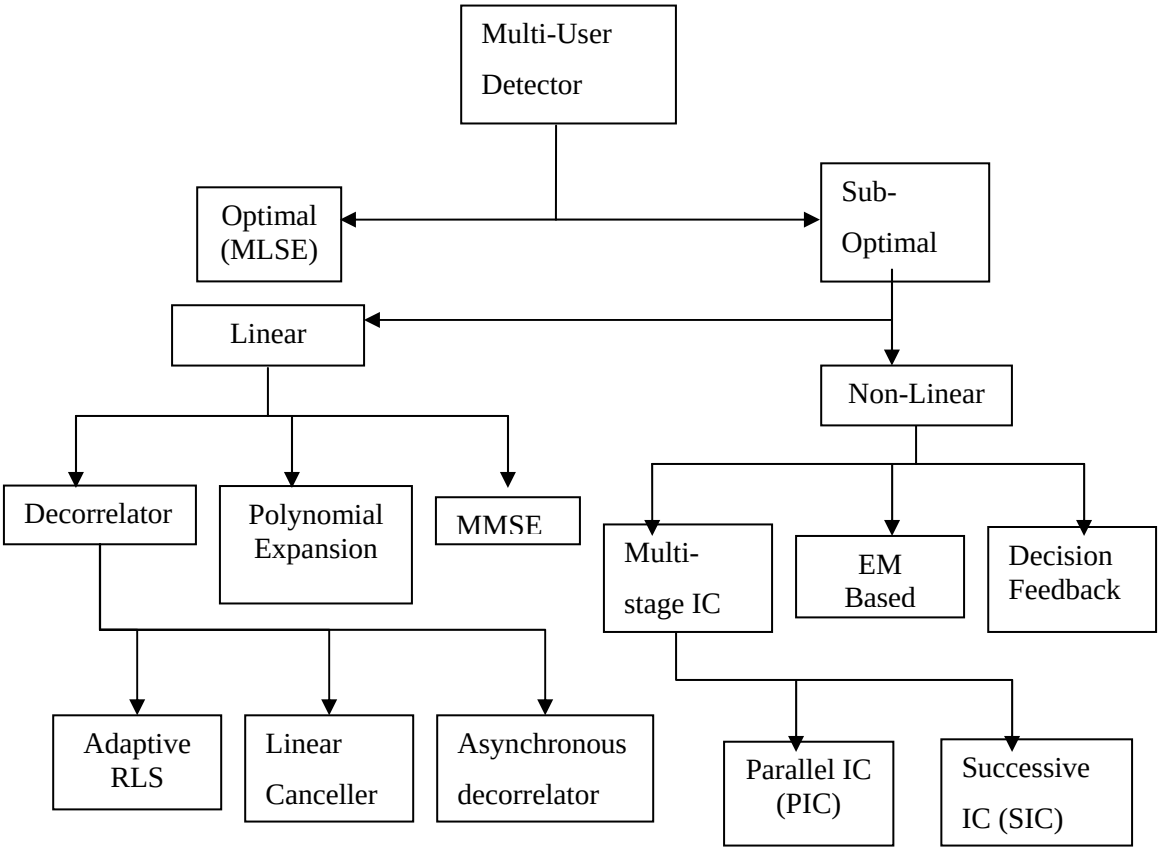

Figure 3.2 Hierarchy of multiuser detectors

Suboptimal multiuser detectors can be classified as *linear* and *non-linear* multiuser detectors. In linear multiuser detectors, a linear transformation is applied to the soft outputs of the conventional detector to produce a new set of decision variables with minimal MAI. The most popular linear multiuser detectors are the *Decorrelating* detector and the *Minimum Mean Square Error* (MMSE) multiuser detector. Non-linear detectors employ subtractive detection, in which the interfering signals are generated and then subtracted from the received signal. Multistage Interference Cancellation schemes are the most popular among the non-linear detectors. Only MMSE multiuser detector is considered in this report.

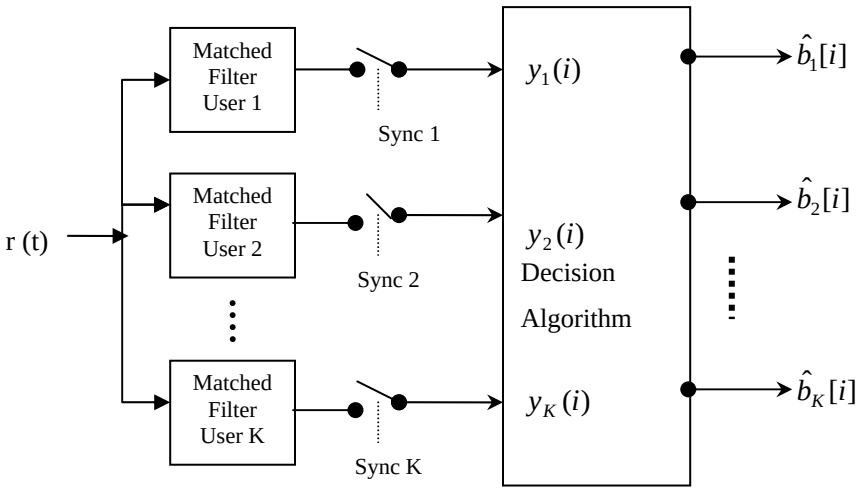

Figure 3.3 Optimal multiuser detector

3.3 MMSE Multiuser Detector

The disadvantage of matched filter is eliminated by the MMSE multiuser detector as it considers the correlative property of MAI. As discussed previously the front end of the multiuser detector is a bank of matched filters. So first let us calculate the output of the matched filter. Again consider the basic synchronous CDMA K-user channel model as input to the matched filter bank (from equation (3.1))

$$r(t) = \sum_{k=1}^{K} A_k\, b_k\, s_k(t) + n(t)$$

then the output y_k of the k^{th} matched filter is

$$y_k = \int_0^T s_k(t)\, r(t)\, dt, \quad k = 1, \ldots, K \tag{3.9}$$

$$= \int_0^T s_k(t) \left[\sum_{j=1}^{K} A_j\, b_j\, s_j(t) + n(t) \right] dt$$

$$= \sum_{j=1}^{K} A_j\, b_j \int_0^T s_j(t)\, s_k(t) + n_k$$

$$= A_k\, b_k + \sum_{j \neq k} A_j\, b_j\, \rho_{jk} + n_k \tag{3.10}$$

where,

$$n_k = \int_0^T n(t)\, s_k(t)\, dt \tag{3.11}$$

$$\rho_{jk} = \int_0^T s_j(t)\, s_k(t)\, dt\,. \tag{3.12}$$

Now putting equation (3.10) in vector form

$$\mathbf{y} = \mathbf{R}\mathbf{A}\mathbf{b} + \mathbf{n} \tag{3.13}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1K} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{K1} & \rho_{K2} & \cdots & \rho_{KK} \end{bmatrix} \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & A_K \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_K \end{bmatrix} \tag{3.14}$$

where, $\mathbf{R}$ is called the normalized cross-correlation matrix and $\mathbf{n}$ is a zero-mean Gaussian random vector with covariance matrix,

$$E[\mathbf{n}\,\mathbf{n}^T] = \sigma^2\, \mathbf{R}\,. \tag{3.15}$$

The MMSE detector implements a linear mapping that minimizes mean square error between the detector output and the bit we are trying to demodulate. So,

$$\hat{\mathbf{w}} = \min_{\mathbf{w}} E\left\{ |\, \mathbf{b} - \mathbf{w}^T\, \mathbf{y}\,|^2 \right\} \tag{3.16}$$

$$\begin{aligned} &= \min_{\mathbf{w}} E[\mathbf{b}\,\mathbf{b}^T] - E[\mathbf{b}\mathbf{y}^T]\mathbf{w}^T - \mathbf{w}\,E[\mathbf{y}\mathbf{b}^T] \\ &\qquad + \mathbf{w}\,E[\mathbf{y}\,\mathbf{y}^T]\mathbf{w}^T\,. \end{aligned} \tag{3.17}$$

Assuming noise and data are uncorrelated, we can say [11]

$$E[\mathbf{b}\mathbf{b}^T] = \mathbf{I}\,, \tag{3.18}$$

$$E[\mathbf{b}\mathbf{y}^T] = E[\mathbf{b}\mathbf{b}^T\,\mathbf{A}\mathbf{R}] = \mathbf{A}\mathbf{R}\,, \tag{3.19}$$

$$E[\mathbf{y}\mathbf{b}^T] = E[\mathbf{R}\mathbf{A}\mathbf{b}\mathbf{b}^T] = \mathbf{R}\mathbf{A}\,, \tag{3.20}$$

$$\begin{aligned} E[\mathbf{y}\,\mathbf{y}^T] &= E[\mathbf{R}\mathbf{A}\mathbf{b}\mathbf{b}^T\,\mathbf{A}\mathbf{R}] + E[\mathbf{n}\mathbf{n}^T] \\ &= \mathbf{R}\mathbf{A}^2\mathbf{R} + \sigma^2\mathbf{R}. \end{aligned} \tag{3.21}$$

Substituting equations (3.18), (3.19), (3.20) and (3.21) into the equation (3.17),

$$E\left\{|\,\mathbf{b}-\mathbf{w}^T\mathbf{y}\,|^2\right\}=\mathbf{I}-\mathbf{A}\mathbf{R}\mathbf{w}^T-\mathbf{w}\mathbf{R}\mathbf{A}+\mathbf{w}\left(\mathbf{R}\mathbf{A}^2\mathbf{R}+\sigma^2\mathbf{R}\right)\mathbf{w}^T\,. \tag{3.22}$$

Equation (3.22) should be minimized according to MMSE criterion, by performing matrix derivative operation on equation (3.22) and equating it to zero,

$$-2\mathbf{A}\mathbf{R}+2\left(\mathbf{R}\mathbf{A}^2\mathbf{R}+\sigma^2\mathbf{R}\right)\mathbf{w}\;=\;0\;. \tag{3.23}$$

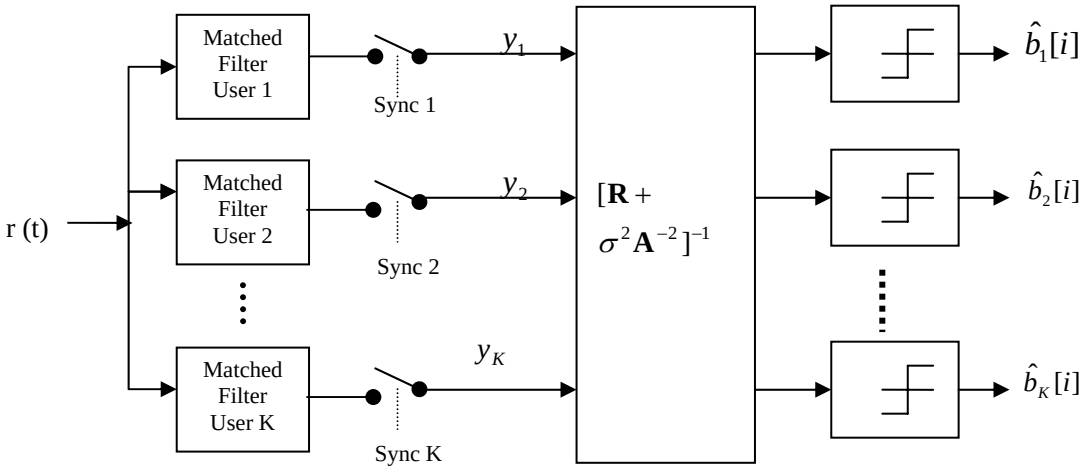

Figure 3.4 MMSE multiuser detector for synchronous CDMA channel [11]

In equation (3.23), assuming the matrix $\mathbf{R}\mathbf{A}^2\mathbf{R}+\sigma^2\mathbf{R}$ to be non-negative definite and $\mathbf{A}$ to be non-singular and rearranging the equation, we obtain

$$\mathbf{w}\;=\mathbf{A}^{-1}[\,\mathbf{R}+\sigma^2\;\mathbf{A}^{-2}\,]^{-1}. \tag{3.24}$$

So the estimate of the k^{th} bit is given by

$$\begin{aligned}\hat{b}_k\;&=sign\left(\frac{1}{A_k}\left([\,\mathbf{R}+\sigma^2\,\mathbf{A}^{-2}\,]^{-1}\,\mathbf{y}\,\right)_k\right)\\&=sign\left(\left([\mathbf{R}+\sigma^2\;\mathbf{A}^{-2}\,]^{-1}\,\mathbf{y}\right)_k\right)\end{aligned} \tag{3.25}$$

here,

$$\sigma^2\,\mathbf{A}^{-2}\;=diag\left\{\frac{\sigma^2}{A_1^2},\;...,\;\frac{\sigma^2}{A_K^2}\right\}. \tag{3.26}$$

Fig 3.5 shows the simulated results for the comparison of matched filter and MMSE multiuser detector in AWGN noise using uncoded BPSK modulation for 20 users using random sequences as spreading signatures. It can be observed that MMSE detector outperforms matched filter.

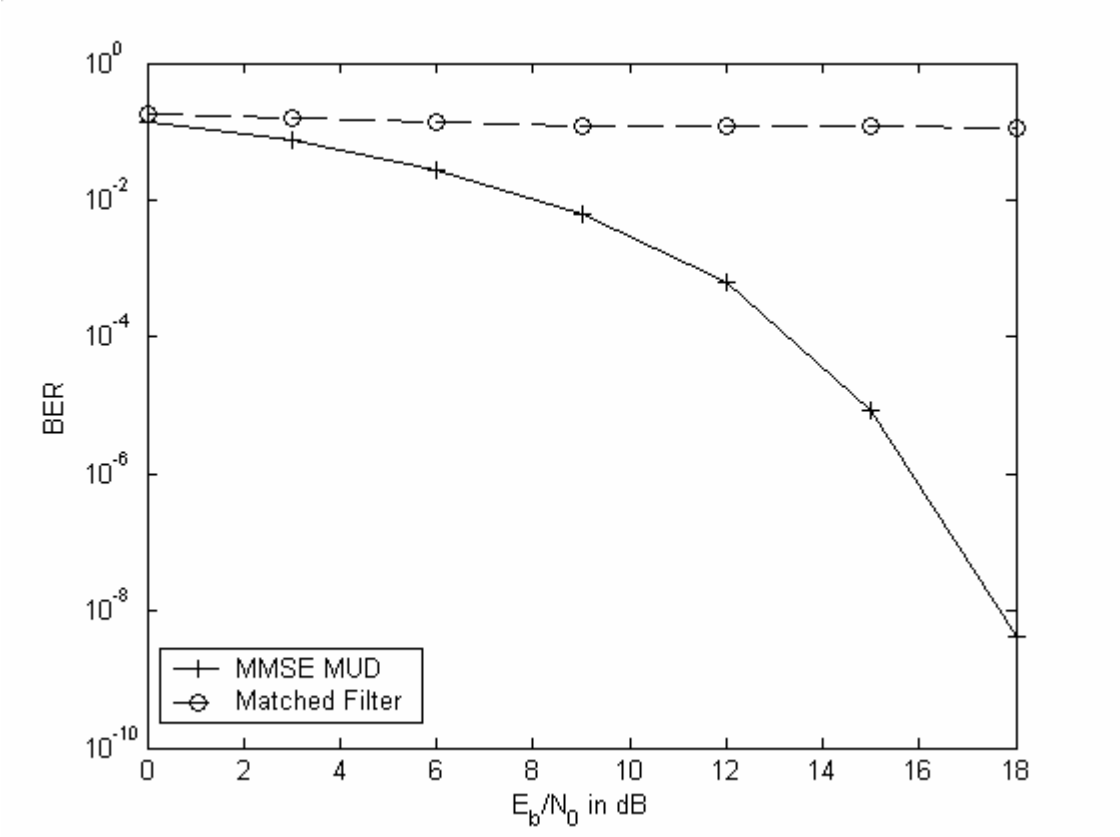

Figure 3.5 Comparison of matched filter and MMSE detector using random sequences for 20 users in AWGN noise with uncoded BPSK modulation

Fig 3.6 shows the simulated results for the comparison of matched filter and MMSE multiuser detector in AWGN noise using uncoded BPSK modulation for 20 users using m-sequences. In this case, matched filter performs better than the previous case that when using random sequences because of the low cross correlation of the m-sequences but not better than the MMSE detector.

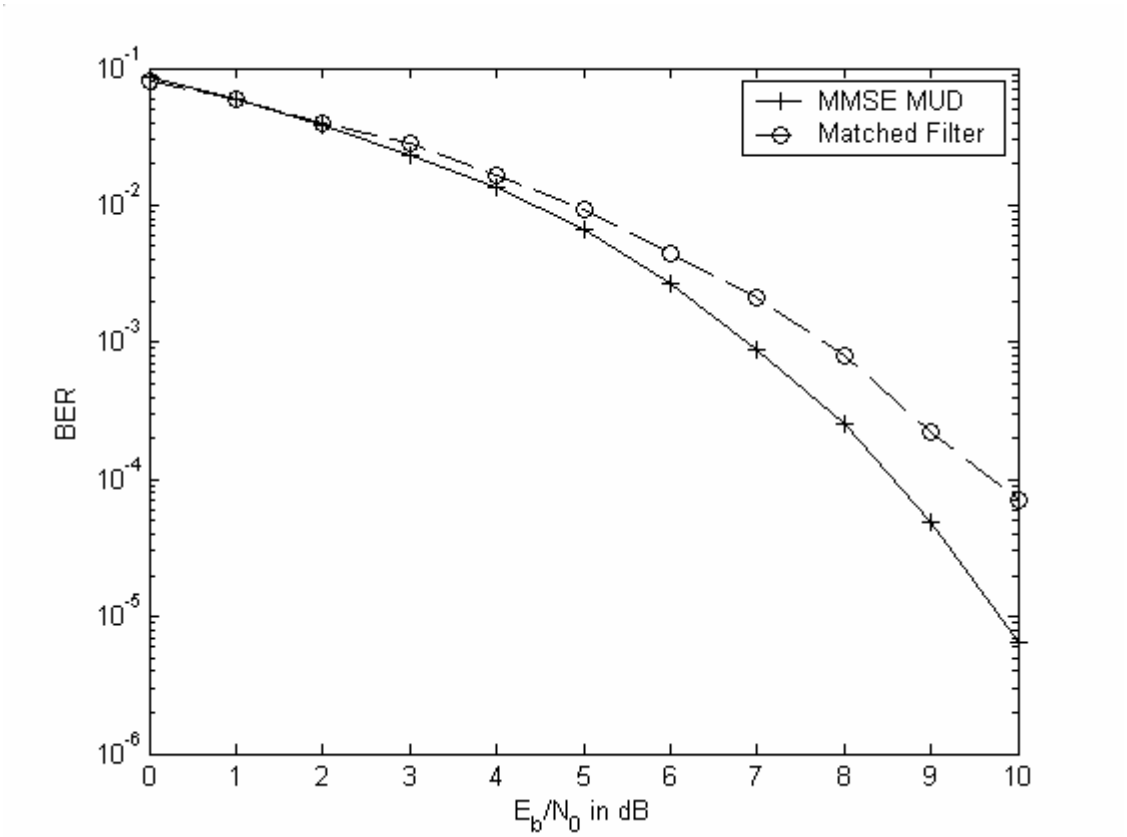

Figure 3.6 Comparison of matched filter and MMSE detector using m-sequences for 20 users in AWGN noise with uncoded BPSK modulation

Chapter 4

Turbo Multiuser Detection

Turbo multiuser detection is defined as "joint channel decoding and multiuser detection using an iterative exchange of soft information between the two processes" in [12]. A considerable amount of research has been focused on this technique. In [13], optimal decoder for asynchronous convolutionally coded CDMA system is developed, but has prohibitive computational complexity of $O(2^{Kl})$, where K is number of users in the channel and l is constraint length of the code. In [14], some low-complexity turbo multiuser receivers are designed. The other problem with this work is that it does not allow the use of a channel interleaver. Another suboptimal turbo multiuser structure is given in [15], but it still has the very high computational complexity of $O(2^K + 2^l)$. A similar approach is presented in [16] also. In this project, a low complexity iterative turbo multiuser receiver for synchronous coded CDMA system is simulated that is developed by Wang and Poor in [1]. This low complexity approximate SISO multiuser detector is based on soft interference cancellation and linear minimum mean square filtering. Furthermore, this project also simulates the iterative multiuser detector of [5], which skips linear MMSE after the soft interference cancellation.

The rest of the chapter is organized as follows. Section 4.1 explains the turbo principle. Section 4.2 presents the system model. Section 4.3 describes the SISO multiuser detector and section 4.4 the SISO channel decoder. Section 4.5 presents the simplified SIC-MF SISO multiuser detector given in [5].

4.1 Turbo Principle

The term "Turbo principle" gained considerable attention after the discovery of the powerful turbo codes [17, 18]. Turbo codes get their name because the decoder uses feedback, like turbo engine. The "turbo" in turbo codes does not apply to the channel codes itself, but it can be related to the iterative way of decoding as in our case. The turbo principle can be applied anywhere where two or more decoders have the information about the data of our interest. This information is iterated between the decoders. The information produced by these decoders should be

uncorrelated at least to start with. As the number of iterations increase, the information will become more correlated and finally there will not be much improvement after certain number of iterations.

At present the turbo principle is applied to many decoding problems such as multiuser detection, serial concatenated decoding, coded modulation, equalization and joint source and channel decoding [19].

In this project the turbo principle is applied between SISO multiuser detector and SISO channel decoder. The decoded information is iterated between these two decoders until the information contained in these two decoders become correlated. Generally in decoding process, this is determined by setting a fixed number of iterations. In this project the information is exchanged between two decoders for five iterations.

The turbo principle gives large performance gain over non-iterative receivers but requires more computational power for iterating data more than once between the decoders.

4.2 System Model

Convolutionally coded K - user synchronous CDMA system is considered in this project. The block diagram of the system is shown in Fig 4.1. The binary information bits d_k for user k, $k = 1, 2, ..., K$, are convolutionally encoded with code rate of R. The convolutionally encoded bits are passed to code bit interleaver. At the receiver, the information is exchanged between the SISO multiuser detector and K - parallel SISO channel decoders using the turbo principle as described in previous section. In this process if one decoder makes an error in decoding, the other decoder receives wrongly decoded code bits and it is not possible to correct these errors resulting loss of information. The interleaver is used to avoid these error bursts. Interleaver achieves this by spreading the channel errors over the codewords in a known way. The convolutionally encoded bits of each user are scrambled to different bit positions using the interleaver. The interleaver used in this system is complete random interleaver without having any structure and each user uses a different interleaver. The interleaver randomly generates one bit position for each coded bit and shifts that coded bit to that particular bit position. Deinterleaver uses these bit positions to rearrange the coded bits in original order. The interleaved coded bits are BPSK symbol mapped and then each data symbol $b_k(i)$ is modulated by spreading waveforms $s_k(t)$. The transmitted signal over AWGN channel is given by [1],

$$r(t) = \sum_{k=1}^{K} A_k \sum_{i=0}^{M-1} b_k(i)\, s_k(t - iT) + n(t) \tag{4.1}$$

where, A_k is amplitude of the k th user's signal

M is the number of symbols per user per frame

$n(t)$ is the zero-mean white Gaussian noise with unit power spectral density

σ^2 is the variance of the noise

The transmitted end of the system is shown in upper part of the Fig 4.1 and receiver end in the bottom half. The receiver performs two successive decisions done by soft-input soft-output (SISO) multiuser detector and a bank of single user SISO channel decoders through iterations using the turbo principle. At each iteration, the extrinsic information delivered by the SISO multiuser detector is calculated and passed to the deinterleaver. The deinterleaved information is sent to the SISO channel decoder as a priori information. The output of the channel decoder is again interleaved, and is fed back to the SISO multiuser detector as a priori data. At the last iteration, the information bits are calculated.

Before proceeding to the receiver structure, let us calculate the output $y_k(i)$ of the k th matched filter in the i th code bit interval [1],

$$y_k(i) = \int_{iT}^{(i+1)T} s_k(t - iT)\, r(t)\, dt, \quad k = 1, \ldots, K. \tag{4.2}$$

Putting it in vector form (refer to equation (3.9)),

$$\mathbf{y}(i) = \mathbf{R}\,\mathbf{A}\,\mathbf{b}(i) + \mathbf{n}. \tag{4.3}$$

The way the receiver structure is simulated in this project is explained in further sections.

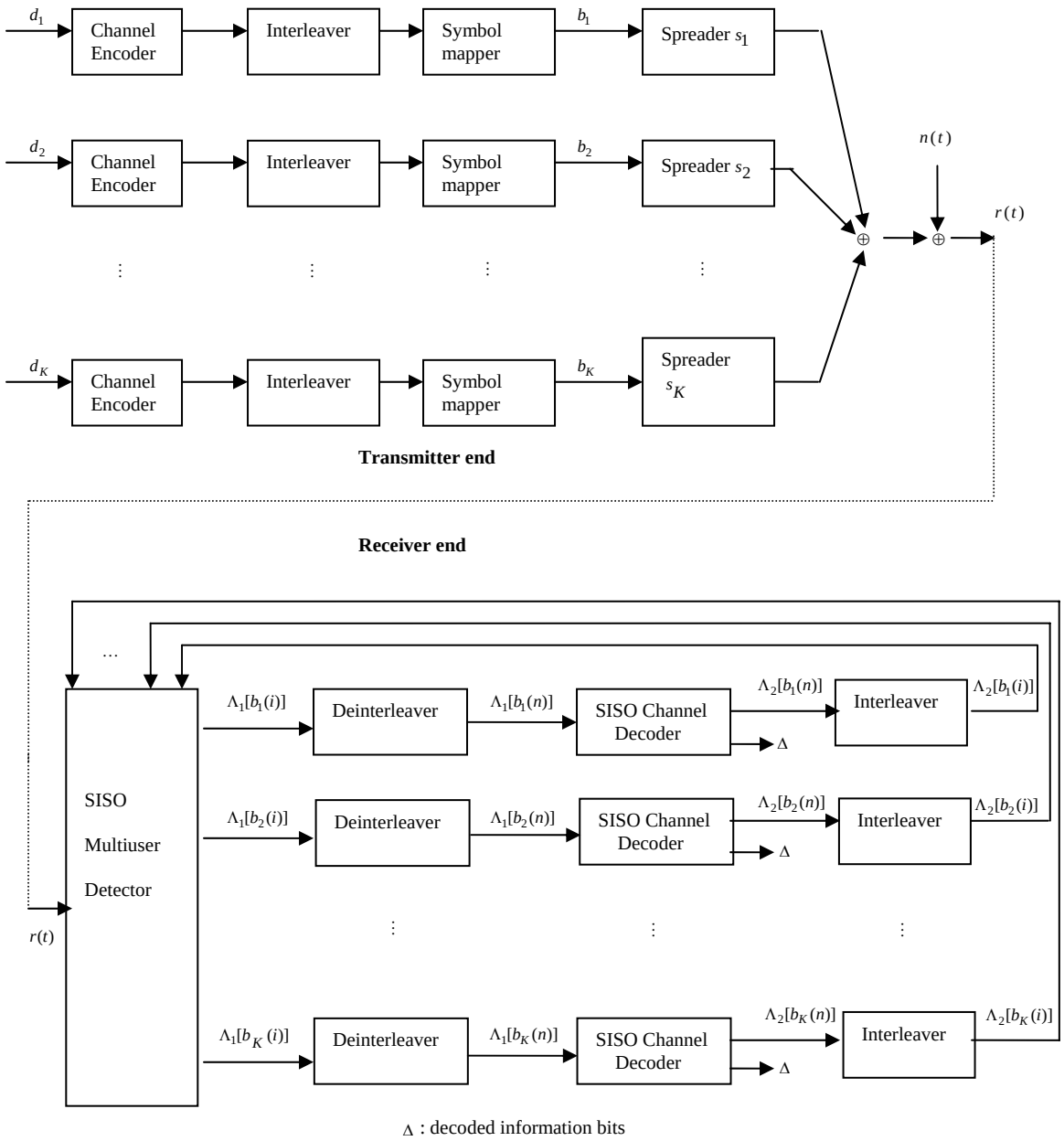

Figure 4.1 Turbo Multiuser Receiver [1]

4.3 SISO Multiuser Detector

SISO multiuser detector is the first part of receiver. The SISO multiuser detector implemented in this project is a low complexity approximate SISO multiuser detector based on soft interference cancellation and linear MMSE filtering developed in [1]. The following derivation for calculating the extrinsic information delivered by SISO multiuser detector is based on this reference.

SISO multiuser detector consists of two stages. First is the soft interference cancellation (SIC) followed by linear MMSE filtering. In soft interference cancellation, we need to calculate the soft estimates of the code bits provided by the SISO channel decoders in the previous iteration. Here it should be noted that, in the first iteration there will be no prior data and hence soft estimates will be zero. From the second iteration onwards, the SISO multiuser detector forms the soft estimates of the code bits provided by the SISO channel decoders in the previous iterations.

Let us define the log-likelihood ratio of the code bits as,

$$\Lambda_2\left[b_j(i)\right] = \log\frac{P[b_j(i)=+1]}{P[b_j(i)=-1]}. \tag{4.4}$$

Exponentiating the equation (4.4) gives,

$$\exp\left(\Lambda_2\left[b_j(i)\right]\right) = \frac{P[b_j(i)=+1]}{P[b_j(i)=-1]}. \tag{4.5}$$

Now the expressions for $P[b_j(i)=+1]$ and $P[b_j(i)=+1]$ in terms of $\exp\left(\Lambda_2[b_j(i)]\right)$ are given by,

$$P[b_j(i)=+1] = \frac{\exp\left(\Lambda_2[b_j(i)]\right)}{1+\exp\left(\Lambda_2[b_j(i)]\right)} \text{ and} \tag{4.6}$$

$$P[b_j(i)=-1] = \frac{1}{1+\exp\left(\Lambda_2[b_j(i)]\right)}. \tag{4.7}$$

From equations (4.6) and (4.7), we can write

$$\begin{aligned} P[b_j] &= P[b_j(i)=b_j], b_j \in \{+1,-1\} \\ &= \frac{\exp\left(b_j\ \Lambda_2[b_j(i)]\right)}{1+\exp\left(b_j\ \Lambda_2[b_j(i)]\right)} \\ &= \frac{\exp\left(b_j\ \Lambda_2[b_j(i)]\right)}{1+\exp\left(b_j\ \Lambda_2[b_j(i)]\right)} \frac{\exp\left(-\frac{1}{2}b_j\ \Lambda_2[b_j(i)]\right)}{\exp\left(-\frac{1}{2}b_j\ \Lambda_2[b_j(i)]\right)} \end{aligned}$$

$$= \frac{\frac{1}{2}(b_j + 1)\exp\left(\frac{1}{2}\Lambda_2[b_j(i)]\right) + \frac{1}{2}(1 - b_j)\exp\left(-\frac{1}{2}\Lambda_2[b_j(i)]\right)}{\exp\left(-\frac{1}{2}\Lambda_2[b_j(i)]\right) + \exp\left(\frac{1}{2}\Lambda_2[b_j(i)]\right)}$$

$$= \frac{\cosh[\frac{1}{2}\Lambda_2[b_j(i)] + \sinh\frac{1}{2}\Lambda_2[b_j(i)]}{2\cosh[\frac{1}{2}\Lambda_2[b_j(i)]}$$

$$= \frac{1}{2}\left[1 + b_j \tanh\left(\frac{1}{2}\Lambda_2[b_j(i)]\right)\right]. \tag{4.8}$$

So the soft estimates of the code bits of all users is

$$\tilde{b}_j(i) = \sum_{b_j \in \{+1,-1\}} b_j P[b_j]$$

$$= \sum_{b_j \in \{+1,-1\}} \frac{b_j}{2}\left[1 + b_j \tanh\left(\frac{1}{2}\Lambda_2[b_j(i)]\right)\right]$$

$$= \frac{1}{2}\left[1 + \tanh\left(\frac{1}{2}\Lambda_2[b_j(i)]\right)\right] - \frac{1}{2}\left[1 - \tanh\left(\frac{1}{2}\Lambda_2[b_j(i)]\right)\right]$$

$$= \tanh\left(\frac{1}{2}\Lambda_2[b_j(i)]\right),\ j = 1, ..., K. \tag{4.9}$$

Define,

$$\tilde{\mathbf{b}}(i) = [\tilde{b}_1(i),\ \tilde{b}_2(i), ..., \tilde{b}_K(i)], \tag{4.10}$$

$$\tilde{\mathbf{b}}_k(i) = \tilde{\mathbf{b}}(i) - \tilde{b}_k(i)\,\mathbf{e}_k$$

$$= [\tilde{b}_1(i), ..., \tilde{b}_{k-1}(i), 0, \tilde{b}_{k+1}(i), ..., \tilde{b}_K(i)]^T \tag{4.11}$$

where $\mathbf{e}_k$ is the vector whose k th element is one and remaining all elements are zeros. Now the soft interference cancellation is performed by subtracting the equation (4.11) from the matched filter output $\mathbf{y}(i)$,

$$\mathbf{y}_k(i) = \mathbf{y}(i) - \mathbf{R}\mathbf{A}\tilde{\mathbf{b}}_k(i)$$

$$= \mathbf{R}\mathbf{A}[\mathbf{b}(i) - \tilde{\mathbf{b}}_k(i)] + \sigma\mathbf{n}(i), k = 1, 2, ..., K. \tag{4.12}$$

The next step is linear minimum mean square error filtering to suppress the remaining interference in soft interference cancellation stage, $\mathbf{y}_k(i)$. The filter $\mathbf{w}_k(i)$ is formed to minimize the mean square error between the code bit $b_k(i)$ and the output of the filter $z_k(i)$. The filter output $z_k(i)$ is expressed as,

$$z_k(i) = \mathbf{w}_k(i)^T \mathbf{y}_k(i). \tag{4.13}$$

Hence the MMSE filter can be defined as,

$$\begin{aligned}\mathbf{w}_k(i) &= \arg\min_{\mathbf{w}\in R^K} E\left\{[b_k(i) - \mathbf{w}^T \mathbf{y}_k(i)]^2\right\} \\ &= \arg\min_{\mathbf{w}\in R^K} \mathbf{w}^T E\left\{\mathbf{y}_k(i)\,\mathbf{y}_k(i)^T\right\}\mathbf{w} - 2\mathbf{w}^T E\left\{b_k(i)\mathbf{y}_k(i)\right\}. \end{aligned}\tag{4.14}$$

The derivation for MMSE filter is similar to the steps followed to derive the MMSE multiuser detector described in section 3.3 of this report except that matched filter output $\mathbf{y}(i)$ is used in equation (3.16) and soft interference cancellation output $\mathbf{y}_k(i)$ is used in equation (4.14). Now

$$E\left\{\mathbf{y}_k(i)\,\mathbf{y}_k(i)^T\right\} = \mathbf{R}\mathbf{A}\,\mathrm{cov}\left\{\mathbf{b}(i) - \tilde{\mathbf{b}}_k(i)\right\}\mathbf{A}\mathbf{R} + \sigma^2\mathbf{R}, \tag{4.15}$$

$$\begin{aligned}E\left\{b_k(i)\,\mathbf{y}_k(i)\right\} &= \mathbf{R}\mathbf{A}E\left\{b_k(i)[\mathbf{b}(i) - \tilde{\mathbf{b}}_k(i)]\right\} \\ &= \mathbf{R}\mathbf{A}\mathbf{e}_k. \end{aligned}\tag{4.16}$$

In equation (4.15), we have

$$\begin{aligned}\mathrm{cov}\left\{\mathbf{b}(i) - \tilde{\mathbf{b}}_k(i)\right\} &= diag[\mathrm{var}\{b_1(i)\},...,\mathrm{var}\{b_{k-1}(i)\},1,\mathrm{var}\{b_{k+1}(i),\},...\mathrm{var}\{b_K(i)\}] \\ &= diag[1-\tilde{b}_1(i)^2,...,1-\tilde{b}_{k-1}(i)^2,1,1-\tilde{b}_{k+1}(i)^2,...,1-\tilde{b}_K(i)^2] \end{aligned}\tag{4.17}$$

since, $$\mathrm{var}\{b_j(i)\} = E\left\{b_j(i)^2\right\} - E\left\{b_j(i)\right\}^2 = 1 - \tilde{b}_j(i)^2. \tag{4.18}$$

Define, $$\mathbf{V}_k(i) = \mathbf{A}\,\mathrm{cov}\left\{\mathbf{b}(i) - \tilde{\mathbf{b}}_k(i)\right\}\mathbf{A} \tag{4.19}$$

$$= \begin{bmatrix} 1-\tilde{b}_1(i)^2 & & & \\ & \ddots & & \mathbf{0} \\ & & 1 & \\ \mathbf{0} & & & \ddots \\ & & & 1-\tilde{b}_K(i)^2 \end{bmatrix}. \tag{4.20}$$

Substituting equations (4.15), (4.16) and (4.19) in equation (4.14), we get

$$\mathbf{w}_k(i) = \mathbf{w}^T[\mathbf{R}\mathbf{V}_k(i)\mathbf{R} + \sigma^2\mathbf{R}]\mathbf{w} - 2\mathbf{w}^T\mathbf{R}\mathbf{A}\mathbf{e}_k. \tag{4.21}$$

Equation (4.21) should be minimized according to MMSE criterion, so performing matrix derivative operation on equation (4.21) and equating it to zero,

$$2\mathbf{w}[\mathbf{R}\mathbf{V}_k(i)\mathbf{R} + \sigma^2\mathbf{R}] - 2\mathbf{R}\mathbf{A}\mathbf{e}_k = 0. \tag{4.22}$$

Therefore the MMSE filter is given by,

$$\mathbf{w}_k(i) = \mathbf{R}\mathbf{A}\mathbf{e}_k[\mathbf{R}\mathbf{V}_k(i)\mathbf{R} + \sigma^2\mathbf{R}]^{-1}$$

$$= \mathbf{A}_k \, \mathbf{R}^{-1} [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k . \tag{4.23}$$

MMSE filter output is obtained by substituting equations (4.12) and (4.23) in equation (4.13),

$$\begin{aligned} z_k(i) &= \mathbf{A}_k \, \mathbf{R}^{-1} [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k [\mathbf{y}(i) - \mathbf{R}\mathbf{A}\tilde{\mathbf{b}}_k(i)] \\ &= \mathbf{A}_k \, \mathbf{e}_k{}^T [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} [\mathbf{R}^{-1}\mathbf{y}(i) - \mathbf{A}\tilde{\mathbf{b}}_k(i)] . \end{aligned} \tag{4.24}$$

In [20], it is shown that the interference and noise in the output of MMSE multiuser detector can be approximated by a Gaussian distribution. So the output of linear MMSE filter, $z_k(i)$ can be expressed as output of an equivalent additive white Gaussian noise channel with input $b_k(i)$,

$$z_k(i) = \mu_k(i) b_k(i) + \eta_k(i) \tag{4.25}$$

where, $\mu_k(i)$ is the amplitude of the k th user's signal,

$\eta_k(i) \sim \mathrm{N}(0, \nu_k^2(i))$ is a Gaussian sample.

Expression for $\mu_k(i)$ can be obtained using equations (4.12) and (4.24),

$$\begin{aligned} \mu_k(i) &= E\{z_k(i) b_k(i)\} \\ &= E\left\{\mathbf{A}_k \, \mathbf{e}_k{}^T [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} [\mathbf{R}^{-1} [\mathbf{R}\mathbf{A}[\mathbf{b}(i) - \tilde{\mathbf{b}}_k(i)] + \sigma \mathbf{n}(i)] - \mathbf{A}\tilde{\mathbf{b}}_k(i)] b_k(i)\right\} \\ &= \mathbf{A}_k \, \mathbf{e}_k{}^T [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} E\left\{b_k(i)\mathbf{A}[\mathbf{b}(i) - \tilde{\mathbf{b}}_k(i)] + b_k(i)\mathbf{R}^{-1}\sigma\mathbf{n}(i)\right\} \\ &= \mathbf{A}_k^2 \, \mathbf{e}_k{}^T [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k \\ &= \mathbf{A}_k^2 [[\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1}]_{kk} \end{aligned} \tag{4.26}$$

where the fourth equality in above equation is obtained using equation (4.16) and also using the fact that the expectancy of noise is zero. The parameter $\nu_k^2(i)$ is derived using equations (4.13), (4.15), (4.19), (4.23) and (4.26),

$$\begin{aligned} \nu_k^2(i) &= \mathrm{var}\{z_k(i)\} = E\{z_k(i)^2\} - \mu_k(i)^2 \\ &= \mathbf{w}_k(i) E\{\mathbf{y}_k(i)\, \mathbf{y}_k(i)^T\} \mathbf{w}_k(i) - \mu_k(i)^2 \\ &= [\mathbf{A}_k \, \mathbf{R}^{-1} [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k^T [\mathbf{R}\mathbf{V}_k(i)\mathbf{R} + \sigma^2 \mathbf{R}] \mathbf{A}_k \, \mathbf{R}^{-1} [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k] \\ &\quad - \mu_k(i)^2 \\ &= [\mathbf{A}_k^2 \, \mathbf{R}^{-1} [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k^T \, \mathbf{R}^2 [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}] \, \mathbf{R}^{-1} [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k] \\ &\quad - \mu_k(i)^2 \\ &= \mathbf{A}_k^2 \, \mathbf{e}_k^T [\mathbf{V}_k(i) + \sigma^2 \mathbf{R}^{-1}]^{-1} \mathbf{e}_k - \mu_k(i)^2 \\ &= \mu_k(i) - \mu_k(i)^2 . \end{aligned} \tag{4.27}$$

Finally the extrinsic information delivered by SIC-MMSE SISO multiuser detector is given by,

$$\Lambda_1[b_k(i)] = \log \frac{P[z_k(i)|\, b_k(i)=+1]}{P[z_k(i)|\, b_k(i)=-1]}$$

$$= \log \frac{\exp\left\{\frac{-[z_k(i)-\mu_k(i)]^2}{2\nu_k^2(i)}\right\}}{\exp\left\{\frac{-[z_k(i)+\mu_k(i)]^2}{2\nu_k^2(i)}\right\}}$$

$$= -\frac{[z_k(i)-\mu_k(i)]^2}{2\nu_k^2(i)} + \frac{[z_k(i)-\mu_k(i)]^2}{2\nu_k^2(i)}$$

$$= \frac{4 z_k(i)\ \mu_k(i)}{2[\mu_k(i)-\mu_k(i)^2]}$$

$$= \frac{2 z_k(i)}{1-\mu_k(i)}. \qquad (4.28)$$

4.4 SISO Channel Decoder

The extrinsic information delivered by SISO multiuser detector is deinterleaved and fed into K-parallel SISO channel decoders as a priori data. So the input to the k th SISO channel decoder are the a priori log-likelihood ratios of the code bits of the k th user. Each SISO channel decoder delivers two outputs. The first output is the LLRs of the code bits. The a posteriori LLR of the code bit b_t^j at the output of the SISO channel decoder is given by,

$$\Lambda_2[b_t^j] = \log \frac{P[b_t^j = +1|decoding]}{P[b_t^j = -1|decoding]}. \qquad (4.29)$$

The second output of the SISO channel decoder is LLRs of the information bits.

$$\Lambda_2[d_t^j] = \log \frac{P[d_t^j = +1|decoding]}{P[d_t^j = -1|decoding]}. \qquad (4.30)$$

Equations (4.29) and (4.30) can be computed using the log-MAP algorithm described in section 2.2.1. The LLR of the information bits, $\Lambda_2[d_t^j]$ can be calculated using the same 3 steps explained at the end of section 2.2.1 that is using the equations from (2.25) to (2.29). Here it should be noted that the LLR of the information bits are only computed at the last iteration. The information bit d_t^j is then obtained by,

$$\hat{d}_t^j = \mathrm{sgn}(\Lambda_2[d_t^j]). \qquad (4.31)$$

The LLR of the code bits $\Lambda_2[b_t^j]$ can also be computed using the same steps with a slight modification in step 3 that is in equation (2.29),

Step 3: Calculation of LLR of each of the code bits $c=0,...,n-1$ of code rate $r=1/n$ for all $i = 0, 1, ..., L-1$ is done according to

$$\Lambda_2[b_t^j] = \max_{S_1^c}{}^*[\overline{\alpha}(j, i) + \overline{\gamma}(s_i \to s_{i+1}) + \overline{\beta}(j', i+1)] - \max_{S_1^c}{}^*[\overline{\alpha}(j, i) + \overline{\gamma}(s_i \to s_{i+1}) + \overline{\beta}(j', i+1)] \quad (4.32)$$

where S_1^c is the set of all state transitions for which the c th code bit is "1" and S_0^c is the set of all state transitions for which the c th code bit is "0".

To obtain a numerically stable algorithm, the parameters $\overline{\alpha}_j$ and $\overline{\beta}_j$ must be scaled as the computation proceeds. The normalization technique to scale the values of $\overline{\alpha}_j$ and $\overline{\beta}_j$ followed in this simulation is different from what is explained in [1]. In this project these values are normalized according to normalization process explained in [21],

$$(\overline{\alpha}_j)_{normalized} = \overline{\alpha}_j - \max_j(\overline{\alpha}_j), \quad (4.33)$$

$$(\overline{\beta}_j)_{normalized} = \overline{\beta}_j - \max_j(\overline{\beta}_j). \quad (4.34)$$

The LLRs of the code bits $\Lambda_2[b_t^j]$ obtained from equation (4.32) are interleaved and fed back to SISO multiuser detector as a priori data as shown in Fig 4.1. At the final iteration, the information bits are obtained from the LLRs of the information bits.

In [1], some recursive procedure is presented to compute the soft output of the SIC-MMSE SISO multiuser detector. This procedure involves two K - vector outer products and two K - vector inner products. So the total computational complexity of this turbo multiuser receiver is $O(K^2 + 2^l)$ which is reasonable for even large K. Hence it can be seen that, the complexity of this turbo multiuser receiver is very less when compared to exponential complexity of other turbo multiuser receivers stated at the starting of this chapter. Computational complexity of those turbo multiuser receivers are exponential in terms of number of users, where as the receiver simulated in this project is not.

4.5 SIC-MF SISO Multiuser Detector

Soft interference cancellation-matched filter SISO multiuser detector is simplification of the SISO multiuser detector explained in section 4.3. In SIC-MF SISO multiuser detector, the linear MMSE filtering step is skipped after the soft interference cancellation stage given in [5]. In SIC-MMSE SISO multiuser detector, linear MMSE filtering is used to suppress the residual interference in soft interference cancellation stage. But in SIC-MF SISO multiuser detector, the residual interference is assumed to be Gaussian. In [22], it is said that this Gaussian approximation gives accurate results for moderate number of users $K(>10)$ when the BER is 10^{-3} or smaller. Therefore the extrinsic information delivered by this detector is given by [5],

$$\Lambda_{SIC\_MF} = \frac{2}{\gamma_k(i)}\left(y_k(i) - \sum_{j \neq k} A_j [\mathbf{R}(i)]_{k,j} \tilde{b}_j(i) \right) \tag{4.35}$$

where, $$\gamma_k(i) = [\mathbf{R}(i)\mathbf{V}_k(i)\mathbf{R}(i) + \sigma^2 \mathbf{R}(i)]_{k,k} - 1. \tag{4.36}$$

In this project, the SIC-MF SISO multiuser is also simulated in similar way as the SIC-MMSE SISO multiuser detector is implemented. That is the receiver performs two successive decisions done by SIC-MF SISO multiuser detector and a bank of single user SISO channel decoders through iterations using the turbo principle. At each iteration, the extrinsic information delivered by the SIC-MF SISO multiuser detector, Λ_{SIC_MF} is calculated and is passed to the deinterleaver. The deinterleaved information is sent to the SISO channel decoder as a priori information. The output of the channel decoder is again interleaved, and is fed back to the SIC-MF SISO multiuser detector as a priori data. At the last iteration, the information bits are calculated.

Next chapter presents the results of the project which illustrate the performance of the turbo multiuser receiver in synchronous coded CDMA system.

Chapter 5

Results

In this chapter, results obtained from the simulation of turbo multiuser receiver in synchronous coded CDMA system given in [1] along with the SIC-MF SISO multiuser detector [5] are presented. The simulation has been developed in MATLAB. This chapter presents three set of curves plotted between SNR in dB and the bit error rate, two of which employ SIC-MMSE SISO multiuser detector in turbo multiuser receiver of four user system with cross correlations of $\rho_{ij} = 0.7$ and $\rho_{ij} = 0.5,$ for $1 \le i, j \le 4$ (here $\rho_{ii} = 1$) respectively and one set of curves employing SIC-MF SISO multiuser detector in turbo multiuser receiver of four user system with cross correlation of $\rho_{ij} = 0.5,$ for $1 \le i, j \le 4$. In all cases the system is simulated one through five iterations.

All users are simulated with convolutional coding with same rate of ½ and constraint length of 5 with generator matrix given by (23, 35 in octal representation),

$$g = \begin{bmatrix} 1\,0\,0\,1\,1 \\ 1\,1\,1\,0\,1 \end{bmatrix}. \tag{5.1}$$

Each user uses a different interleaver generated randomly and same set of interleavers are used for all simulations. The frame size of information bits used in this simulation is 128. All the four users have the same power. In all of the three curves, the single user performance (that is $\rho_{ij} = 0$,) simulated using log-MAP algorithm is also presented to assess the performance of turbo multiuser receiver. Fig 5.1 shows the performance of turbo multiuser receiver employing SIC-MMSE multiuser detector with four users with equal power and cross correlation of $\rho_{ij} = 0.7,$ for $1 \le i, j \le 4$ in AWGN noise using BPSK modulation. From Fig 5.1, it can be said that even though the cross correlation between the users is high, $\rho = 0.7,$ the performance of turbo multiuser receiver still approached the single user performance after the fifth iteration. Note that the performance of this iterative receiver after the first iteration corresponds to the performance of a traditional non-iterative receiver.

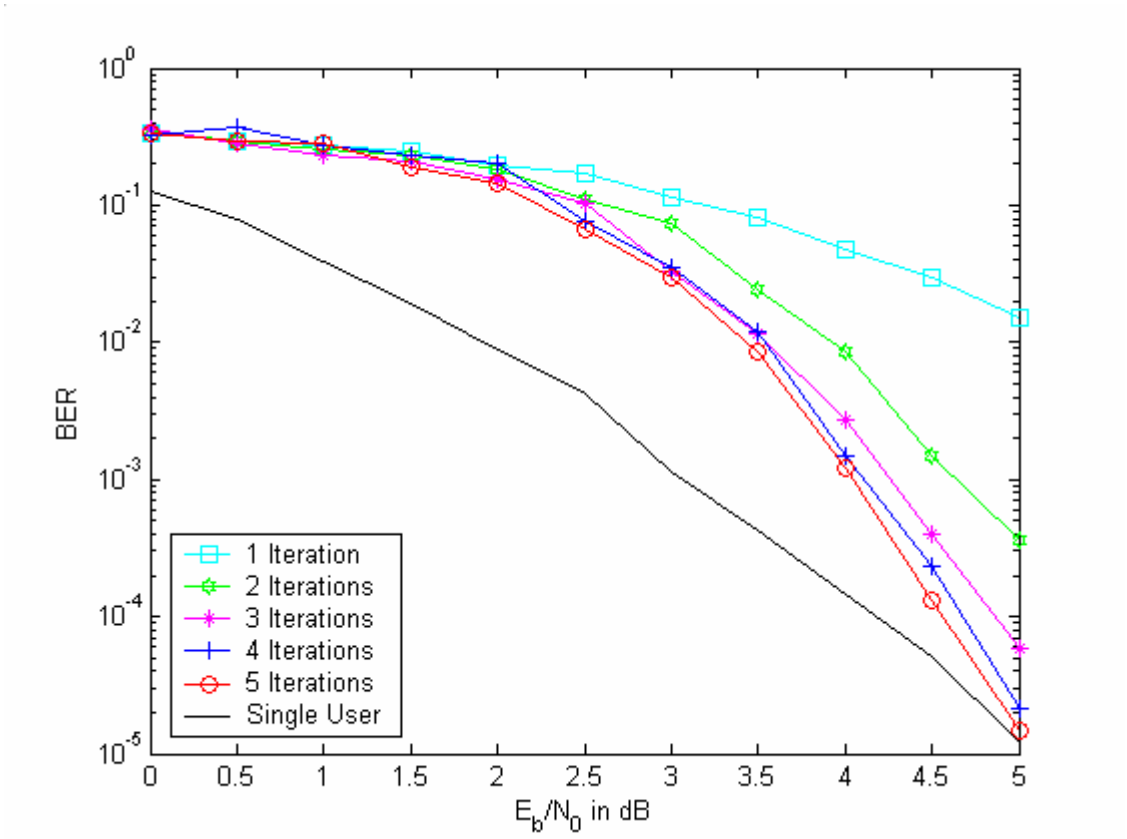

Figure 5.1 Performance of Turbo multiuser receiver employing SIC-MMSE multiuser detector with $K=4$ *and* $\rho_{ij}=0.7$*, rate ½ convolutional code in AWGN noise using BPSK modulation. All users have equal power.*

Fig 5.2 shows the performance of turbo multiuser receiver employing SIC-MMSE multiuser detector with four users with equal power and cross correlation of $\rho_{ij}=0.5,$ for $1\le i, j\le 4$ in AWGN noise using BPSK modulation. From the Fig 5.2, it can be seen that at reasonably high SNR, the performance of low complexity turbo multiuser receiver approaches single user performance after second iteration only with relatively low cross correlation between the users, $\rho=0.5$.

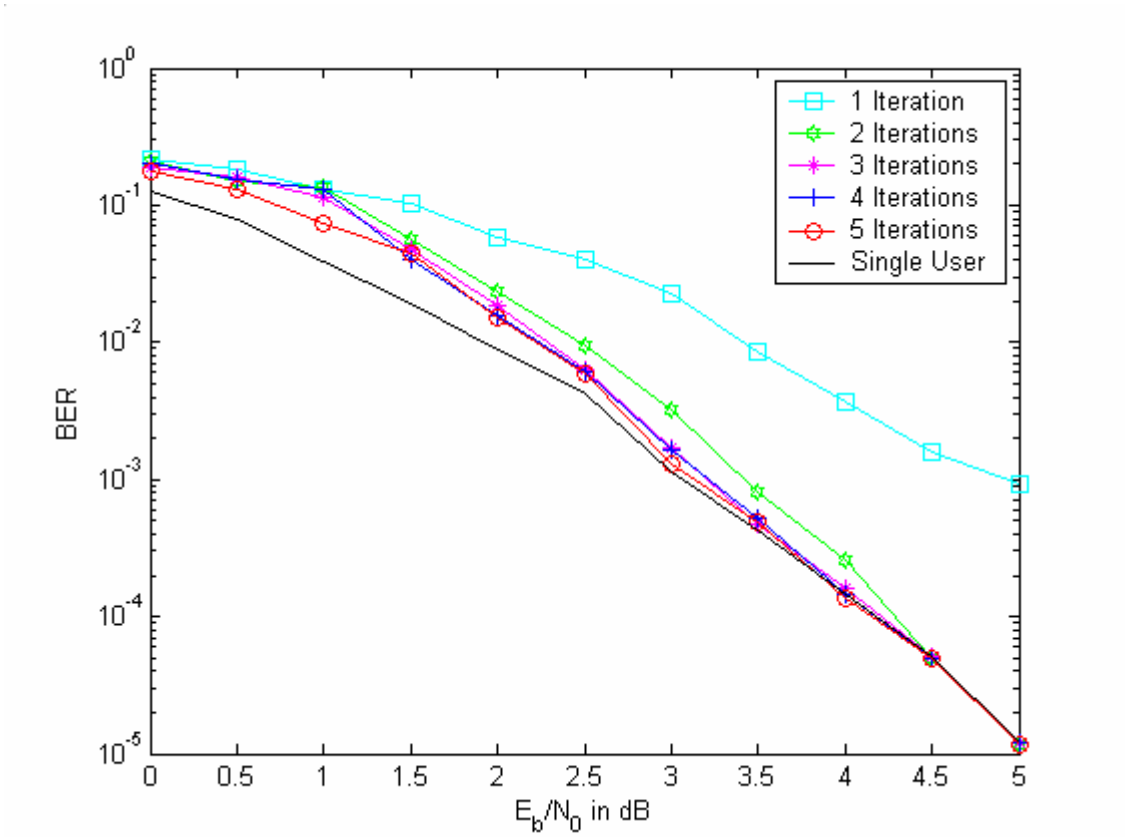

Figure 5.2 Performance of Turbo multiuser receiver employing SIC-MMSE multiuser detector with $K=4$ *and* $\rho_{ij}=0.5$, *rate ½ convolutional code in AWGN noise using BPSK modulation. All users have equal power.*

Fig 5.3 shows the performance of turbo multiuser receiver employing SIC-MF multiuser detector with four users with equal power and cross correlation of $\rho_{ij}=0.5,$ for $1\le i, j\le 4$ in AWGN noise using BPSK modulation. This plot shows the effect of the iterative structure. It can be seen that even after skipping the linear MMSE filtering, the performance approaches single user performance after the fifth iteration with relatively low cross correlation between the users.

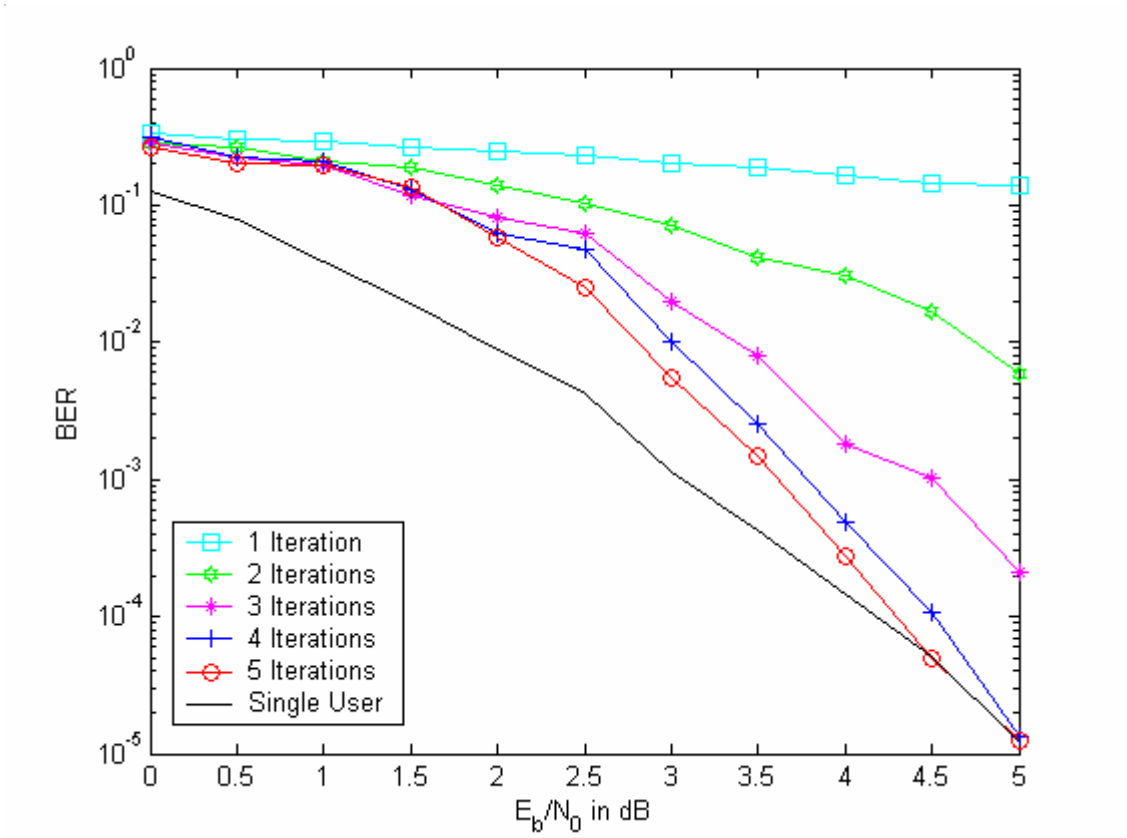

Figure 5.3 Performance of Turbo multiuser receiver employing SIC-MF multiuser detector with $K=4$ *and* $\rho_{ij}=0.5$ *, rate ½ convolutional code in AWGN noise using BPSK modulation. All users have equal power.*

When compared the performances of the SIC-MMSE SISO multiuser detector and SIC-MF SISO multiuser detector with equal cross correlation of $\rho=0.5$ from Fig 5.2 and Fig 5.3, the SIC-MMSE SISO multiuser detector performs better than other one because it suppresses the residual interference in the channel using linear MMSE filtering.

It can be observed from all the three curves that the performance of turbo multiuser receiver converges toward the single user performance at higher SNR as the number of iterations increases.

Chapter 6

Conclusions

In this project, the turbo multiuser receiver for synchronous coded CDMA system is simulated in MATLAB and the performance curves are presented in previous chapter. In particular, the performance of two turbo multiuser receivers, one employing SIC-MMSE SISO multiuser detection and other employing SIC-MF SISO multiuser detection are compared. From all of the three curves presented in Chapter 5, it can be observed that at high signal to noise ratios, the effects of MAI in the channel can be completely eliminated by the iterative procedure, approaching the single user performance.

Note that at first iteration, the outputs of SISO multiuser detector and SISO channel decoders are statistically independent. After certain number of iterations, they exchange the same information indirectly, so the information will become more and more correlated and finally the improvement through iterations will diminish.

Possible extensions for this project might be:

- SISO multiuser detector for multipath channel
- Application of turbo multiuser technique to relay networks

References

[1] X. Wang and H.V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, No. 7, pp.1046–1061, July 1999.

[2] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379-423 and 623-656, July and Oct. 1948.

[3] T.S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., Upper Saddle River, NJ: Prentic Hall PTR, 2002, pp. 456-459.

[4] J.S. Lee and L.E. Miller, *CDMA Systems Engineering Handbook*, Artech House, 1998, pp.18-19, 543-547.

[5] G. Yue, X. Wang and K.R. Narayanan, "Design of low density parity check codes for turbo multiuser detection," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, vol. 4, pp. 2703-2707, May 2003.

[6] M. C. Valenti. (July 1999). Dissertation: *Iterative Detection and Decoding for Wireless Communications.* [Online]. Available: http://www.csee.wvu.edu/~mvalenti/documents/valenti1999a.pdf

[7] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268-278, Mar. 1973.

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

[9] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc., IEEE GLOBECOM*, pp. 1680-1686, 1989.

[10] P. Robertson, P. Hoeher, and E. Villeburn, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. on Telecommun.*, vol.8, pp. 119-125, Mar. /Apr. 1997.

[11] S. Verdu, *Multiuser Detection*, Cambridge University Press, 1998.

[12] H. V. Poor, "Turbo multiuser detection: An overview," *IEEE 6th Int. Symp. on Spread-Spectrum Tech. & Appli.*, NJIT, New Jersey, USA, Sept. 6-8, 2000.

[13] T. R. Giallorenzi and S.G. Wilson, "Multiuser ML sequence estimator for convolutional coded asynchronous DS-CDMA systems," *IEEE Trans. Commun.*, vol. COM-44, pp. 997-1008, Aug. 1996.

[14] T. R. Giallorenzi and S.G. Wilson, "Suboptimal multiuser receivers for convolutionally coded asynchronous DS-CDMA systems," *IEEE Trans. Commun.*, vol. COM-44, no. 9, pp. 1183-1196, Sept. 1996.

[15] M. Moher, "An iterative multiuser decoder for near-capacity communications," *IEEE Trans. Commun.*, vol. 46, pp. 870–880, July 1998.

[16] M. C. Reed, "Iterative multiuser detection for CDMA with FEC: Near single user performance," *IEEE Trans. Commun.*, vol. 46, pp. 1693–1699, Dec. 1998.

[17] C. Berrou and A. Glavieux, "Near optimum error-correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, Oct.1996.

[18] C. Berrou, A. Glavieux and P. Thitimajshhima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. 1993 Int. Conf. on Communications (ICC'93)*, 1993, pp. 1064-1070.

[19] J. Hagenauer, "The Turbo principle: Tutorial introduction and state of the art," in *Proc. International Symposium on Turbo codes and Related Topics*, Brest, France, Sept. 1997, pp. 1-11.

[20] H. V. Poor and S. Verdu, "Probability of error in MMSE multiuser detection," *IEEE Trans., Inform. Theory*, vol. IT-43, pp. 858-871, May 1997.

[21] M. C. Valenti and J. Sun, "The UMTS turbo code and an efficient decoder implementation suitable for software-defined radios," *International Journal of Wireless Information Networks*, vol. 8, no. 4, Oct. 2001.

[22] T. Eng and L. B. Milstein, "Coherent DS-CDMA performance in Nakagami multipath fading," *IEEE Trans. Commun.*, vol.43, no. 2/3/4, pp. 1134-1143, Feb./Mar./April 1995.