ITERATIVE LOW-COMPLEXITY MULTIUSER DETECTION
AND DECODING FOR CODED UWB SYSTEMS

A Thesis by

Arun D. Sathish

B.E. in Electronics & Communications Engineering

Visveswaraih Technological University, India, 2002

Submitted to the Department of Electrical and Computer Engineering
and the faculty of the Graduate School of
Wichita State University in partial fulfillment of
the requirements for the degree of
Master of Science

July 2006
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
I have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Hyuck Kwon, Committee Chair

We have read this thesis and recommend its acceptance:

Edwin Sawan, Committee Member

Kurt Soschinske, Committee Member
DEDICATION

To my beloved parents

and

(Sri Dharmasthala Manjunatha Swamy)
ACKNOWLEDGEMENTS

To All Those Who Made This Possible: Without your help, dedication and guidance, I would not have been able to complete this thesis. You have played indispensable roles in my life and work. I thank you from my heart:

Foremost, thank you, Dr. Sudharman K. Jayaweera. You taught me what research is. Your knowledge, thinking, and ideas guided my work in the right direction. Dr. Jayaweera, thank you also for being available whenever I knocked on your door. You have been invaluable to me.

Dearest Dr. M. Edwin Sawan, my work supervisor. I will never forget you — nor your steady guidance and advice through thick and thin during my work. More than that, however, I thank you for being a true human being and real friend. I can only hope to be half as blessed with a supervisor such as you someday in my future. My utmost thanks to you, eminent Dr. Sawan.

Dr. Hyuck M. Kwon, I am indebted to you, also — especially for your help with my thesis completion at the last minute. My sincerest thanks, also, for sharing your concepts, suggestions, and encouragement for my future work based on this thesis.

Dr. Kurt Soschinske, a special thanks to you for giving your time as a committee member in my oral presentation — especially for your patience in understanding.

My loving parents in India, without whom nothing would have been possible for me. Father and Mother, you taught me the value of hard work by your own example. I share this moment of happiness with you, most of all. You have supported me enormously during the entire tenure of my Master's degree. I love you.

Thank you, too, Mr. Martin G. Connolly, my friend and Journalist in Kansas City, Missouri, for your outgoing interest, friendship and patience, listening whenever I babbled

iv
on the phone with you.

To everyone else behind me — you know who you are: my relatives, friends, roommates, the Graduate School, and the College of Engineering at Wichita State University. You all have helped me in your own special ways.

Finally, in confession, thanks to two other "friends" — though both dubious: Mr. Marlboro and Mr. Johnny Walker. They have helped relieve some of the stress at times. I hope to quit the former and moderate the latter.

Thank you all so much, again.
ABSTRACT

In general, ultra wideband (UWB) signals are transmitted using very short pulses in time domain, thus promising very high data rates. In this thesis, a receiver structure is proposed for decoding multiuser information data in a convolutionally coded UWB system. The proposed iterative receiver has three stages: a pulse detector, a symbol detector, and a channel decoder. Each of these stages outputs soft values, which are used as \textit{a priori} information in the next iteration. Simulation results show that the proposed system can provide performance very close to a single-user system.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION AND PREVIEW</td>
</tr>
<tr>
<td>1.1</td>
<td>Definition of Ultra Wideband</td>
</tr>
<tr>
<td>1.1.1</td>
<td>Energy Bandwidth Concept</td>
</tr>
<tr>
<td>1.2</td>
<td>Literature Review</td>
</tr>
<tr>
<td>1.3</td>
<td>Thesis Contribution</td>
</tr>
<tr>
<td>1.4</td>
<td>Thesis Outline</td>
</tr>
<tr>
<td>2.</td>
<td>TURBO MULTIUSER DETECTOR</td>
</tr>
<tr>
<td>2.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2.2</td>
<td>Signal Model and System Description</td>
</tr>
<tr>
<td>2.3</td>
<td>Turbo-Principled Multiuser Detector</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Pulse Detector</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Symbol Detector</td>
</tr>
<tr>
<td>3.</td>
<td>CONVOLUTIONAL CODING AND DECODING</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Convolutional Encoding</td>
</tr>
<tr>
<td>3.1.2</td>
<td>States and Trellis Diagram</td>
</tr>
<tr>
<td>3.2</td>
<td>The Maximum A-Posteriori Algorithm</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Mathematics Preliminaries</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Calculation of the $\alpha_{t}(s)$ Values</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Calculation of the $\beta_{t}(s)$ Values</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Calculation of the $\gamma_{t}(s)$ Values</td>
</tr>
<tr>
<td>4.</td>
<td>PROPOSED ITERATIVE RECEIVER MODEL</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>4.2</td>
<td>Proposed Model</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Pulse Detector</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Symbol Detector</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Channel Decoder</td>
</tr>
<tr>
<td>5.</td>
<td>PERFORMANCE ANALYSIS</td>
</tr>
<tr>
<td>6.</td>
<td>CONCLUSIONS AND FUTURE WORK</td>
</tr>
<tr>
<td>5.1</td>
<td>Conclusions</td>
</tr>
<tr>
<td>5.2</td>
<td>Future Work</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

REFERENCES ................................................................................................................................................29
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Energy bandwidth</td>
</tr>
<tr>
<td>1.2</td>
<td>A UWB signal</td>
</tr>
<tr>
<td>1.3</td>
<td>Non-UWB Signal</td>
</tr>
<tr>
<td>2.1</td>
<td>Turbo-principled multiuser detector</td>
</tr>
<tr>
<td>3.1</td>
<td>Convolutional encoder CC (2, 1, 2)</td>
</tr>
<tr>
<td>3.2</td>
<td>Input and output bits of CC(2, 1, 2)</td>
</tr>
<tr>
<td>3.3</td>
<td>RSC encoder</td>
</tr>
<tr>
<td>3.4</td>
<td>Possible state transition in RSC component code</td>
</tr>
<tr>
<td>3.5</td>
<td>MAP decoder trellis</td>
</tr>
<tr>
<td>4.1</td>
<td>Coded UWB system</td>
</tr>
<tr>
<td>5.1</td>
<td>Average probability of turbo UWB receiver with $N_c = 20$, $N_r = 10$, and $K = 20$</td>
</tr>
<tr>
<td>5.2</td>
<td>Average probability of turbo UWB receiver with $N_c = 20$, $N_r = 10$, and $K = 10$</td>
</tr>
<tr>
<td>5.3</td>
<td>Average probability of turbo UWB receiver with $N_c = 10$, $N_r = 10$, and $K = 10$</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction and Preview

1.1 Definition of Ultra Wideband

Ultra wideband (UWB) signals are very short pulses of a few hundred picoseconds duration. These systems have a very large bandwidth and hence are called ultra wideband.

The United States Federal Communications Commission (FCC) defines a UWB system as any radio system with a fractional bandwidth greater than 20 percent of the center frequency or a -10dB bandwidth greater than 500MHz.

1.1.1 Energy Bandwidth Concept

Figure 1.1 illustrates the energy bandwidth concept with an example. Here, $f_l$ is the lower limit of energy spectral density, $f_h$ is the higher limit of energy spectral density, $\frac{f_h+f_l}{2}$ is the center frequency of the spectrum, and $\frac{(f_h-f_l)}{f_h+f_l}$ is the fractional bandwidth.

![Energy Bandwidth](image)

Figure 1.1: Energy bandwidth.

If fractional bandwidth is greater than 0.20, then the signal is UWB. The traditional
concepts in radio communication is that a signal is wideband or broadband (vs. narrowband) when its bandwidth is large. But in the case of UWB, due to the absence of modulating carrier frequency, reference is made to the center frequency of the spectrum. Figure 1.2 represents a UWB signal where the fractional bandwidth is greater than 0.20, and Figure 1.3 is a non-UWB signal with fractional bandwidth less than 0.20.

Figure 1.2: UWB signal

Figure 1.3: Non-UWB signal
1.2 Literature Review

The radiation of ultra wideband signals in wireless communication is by very short pulses of a few hundred picoseconds duration. Hence, these systems have a very large bandwidth. UWB systems offer many advantages over narrow-band or conventional wide-band systems: simple transceiver designs, good anti-jam capabilities, low probability of detection, and reduced fading margins [1]. Many signaling methods are used for transmitting over UWB channels. Any existing spread spectrum technique can be used for transmitting over UWB channels [2], [3], [4]. The actual use of conventional spread-spectrum methods for transmitting over UWB channels will encounter some technical difficulties. For example, in the UWB system, implementation of even a simple matched filter detector requires a chip rate of 7.5 GHZ, a high frequency that is almost impossible with today’s sophisticated technology. To overcome these difficulties, time-hopping impulse radios (TH-IRs) are proposed as a modulation scheme for UWB systems [5], [6]. TH-IRs have drawn considerable attention among both researchers and practitioners over the past few years because of their advantages over the UWB system. For example, the matched filter detector needs to sample the received signal only at the instance when a pulse of user of interest arrives at the receiver. In TH-IR systems, a train of short pulses is transmitted, and the information is conveyed by the polarity of the transmitted pulses. By decreasing the transmitted pulse width, the transmitted bandwidth can be increased up to the point where the system becomes a UWB system. By using very short pulses, the transmitted energy is spread over a very large bandwidth, saving the need for spreading sequences. In addition, in order to allow many users to share the same channel, an additional time shift at the beginning of each pulse, which is known to the receiver, helps to avoid the catastrophic collision between two users on the same channel. It has been observed in [7], [8], [9] that the transmitted and received signals of TH-IR systems can be described by the same models used for the transmitted and received signals of direct-sequence code division multiple access (DS-CDMA) systems. It has already been noted that almost all multiuser detectors designed for DS-CDMA systems can be used
in a TH-IR system, with or without slight modifications [9].

1.3 Thesis Contribution

This thesis has proposed a iterative receiver with three stages: a pulse detector, a symbol detector, and a channel decoder. The first two stages are the low complexity turbo principled multiuser detector canceling the interference among users. The third stage is a channel decoder, which corrects errors in the received bits due to transmission noise. It is shown that this proposed receiver structure performs very close to a single user bound at high signal-to-ratio (SNR).

This work was published in the 9th International Symposium on Spread Spectrum Techniques and Applications (ISSSTA 2006) held at Manaus, Brazil, under the same title “Iterative Low-Complexity Multiuser Detection and Decoding for Coded UWB Systems.”

1.4 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 describes both continuous-time and discrete-time signals for the transmitted and received signals in TH-IR systems and discusses the turbo-principled multiuser detector. Chapter 3 provides a brief introduction to convolutional coding and decoding. Chapter 4 presents a proposed iterative multi-user receiver structure, and Chapter 5 explains the simulation results. Finally Chapter 6 presents the conclusions and future work.
Chapter 2

Turbo Multiuser Detector

2.1 Introduction

This chapter we discusses the multiuser detection part of the proposed receiver structure. The multiuser detector is essentially the same as the turbo-principled multiuser detector [10]. Turbo-decoding mud [11] has two stages as most of the turbo-decoding algorithm. Each stage exchanges the extrinsic information with the other and thus provides an efficient way to estimate the information symbol. This multiuser detector does not have actual decoding of error like other turbo-decoding algorithms. Since two stages exchange the extrinsic information, it is called a “turbo” multiuser detector.

2.2 Signal Model and System Description

The transmitted signal, of say the $k^{th}$ user, in a random TH-IR system is described by the following general model [10]:

$$s_{tr}^k(t) = \sum_{j=-\infty}^{\infty} b^j_k \chi_j(t - jT_f - c^j_T)$$  \hspace{1cm} (2.2.1)

where $w_{tr}(t)$ is the transmitted UWB pulse; $\{b^j_k\}$ is the binary sequence of information symbols transmitted by the $k^{th}$ user, where $b^j_k \in \{+1, -1\}$; $\{c^j_T\}$ is a pseudorandom time hopping sequence of the $k^{th}$ user taking values in $\{0, 1, \ldots, N_c - 1\}$; and $N_c$ is the number
of chips in which a pulse can take its position. These \( \{ c^k_j \} \) provide an additional shift of \( c^k_j T_c \) seconds to \( j \)th pulse of the \( k \)th user. \( T_f \) is the nominal pulse repetition time. \( N_f \) is the number of pulses used to transmit one information symbol, and \( \lfloor \cdot \rfloor \) denotes the integer part.

The received signal at the antenna output is

\[
r(t) = \sum_{k=1}^{K} A_k \sum_{j=-\infty}^{\infty} b^k_j(t) w_{r,s}(t - j T_f - c^k_j T_c) + n(t)
\]

where \( w_{r,s}(t) \) is the received pulse, \( n(t) \) is the additive noise, and \( A_k \) is the received amplitude of the signal of the \( k \)th user. The received signal equation (2.2.2) is passed through a linear filter matched to \( w_{r,s}(t) \), and the output of this filter is sampled every \( T_c \) seconds [10]. A sufficient statistic for detecting the \( i \)th information symbols of all users is given by \( r[i] \), obtained by stacking up all the samples corresponding to the \( i \)th frame:

\[
r[i] = S[i] A b[i] + n[i]
\]

where \( S_{N_r \times N_f \times K} \) is a matrix whose non-zero elements in the \( k \)th column are placed at indices representing the time instances where pulses from the \( k \)th user are received [10]. \( A_{K \times K} = \text{diag}(A_1, \ldots, A_K) \) is a diagonal matrix with the gains between the transmitter and the receiver on its diagonal, and \( b[i] \) is a \( K \)-array vector whose \( k \)th element is the \( i \)th information symbol transmitted by the \( k \)th user. The receiver noise is assumed to be white and Gaussian, and \( n[i] \) is a zero mean Gaussian random vector with correlation matrix \( \sigma_n^2 I \), where \( \sigma_n^2 = \frac{N_r}{2} \). If \( l(j, k) \) is the time index at which the \( j \)th pulse from the \( k \)th user was transmitted, then from equation (2.2.3), \( r_{l(j,k)} \) obeys the following model [10]:

\[
r_{l(j,k)} = 1_{K_f^k+1} A^k b^k + n_{l(j,k)}
\]

where the information symbols colliding with the \( j \)th pulse of the \( k \)th user are denoted by the vector \( b^k_j = \left[ b^k_{j,1} f^k_j(1), \ldots, b^k_{j,1} f^k_j(K^k_{f,j}) \right] \). The \( j \)th pulse of the \( k \)th user. \( K^k_{f,j} \) is the number of users colliding with the \( j \)th pulse of the \( k \)th user. \( 1_{K_f^k+1} \) is the \( 1 \times (K_f^k + 1) \) vector of all ones, the diagonal matrix \( A^k = \text{diag} \left( A_k, A_{f^k_j(1)}, \ldots, A_{f^k_j(K^k_{f,1})} \right) \) contains the amplitudes of the pulses received from the \( k \)th user and the users colliding with
the $j^{th}$ pulse of the $k^{th}$ user, and $n_{(j,k)}$ is the additive white Gaussian noise with zero mean and $\sigma_n^2 = N_0$.

### 2.3 Turbo-Principled Multiuser Detector

The general structure of this multiuser detector is very similar to other turbo-like algorithms. It is composed of two stages: a pulse detector and a symbol detector as shown in Figure 4.1. In the first stage, it is assumed that different pulses from the same user correspond to an independent information symbol. In the second stage, information that all the pulse from the same user correspond to the same symbol is exploited [10].

![Turbo-principled multiuser detector](image)

Figure 2.1: Turbo-principled multiuser detector.

Let $b_j^k$ be the information symbol transmitted by the $j^{th}$ pulse of the $k^{th}$ user. It will be shown that at the $n^{th}$ iteration, the pulse detector computes the *a posteriori* log-likelihood ratio (LLR) of the pulse $b_j^k$, denoted as $\Lambda_1^n(b_j^k)$, given the received signal, the other users information about the transmitted bits and the *a priori* LLR $b_j^k$ obtained from the symbol detector as

$$\Lambda_1^n(b_j^k) = \lambda_1^n(b_j^k) + \Lambda_2^{n-1}(b_j^k)$$  \hspace{1cm} (2.3.1)

where $\lambda_1^n(b_j^k)$ is the extrinsic information, and $\Lambda_2^{n-1}(b_j^k)$ represents the *a priori* LLR of $b_j^k$, which is computed by the symbol detector at the $(n-1)^{th}$ iteration.

The symbol detector computes the *a posteriori* LLR of $b_j^k$, denoted by $\Lambda_2^n(b_j^k)$, given the
soft information $\lambda_1^n(b_j^k)$ from the pulse detector as

$$\lambda_2^n(b_j^k) = \lambda_2^n(b_j^k) + \lambda_1^{n-1}(b_j^k)$$  \hspace{1cm} (2.3.2)

where $\lambda_2^n(b_j^k)$ is the extrinsic information, which is fed into a pulse detector and $\lambda_1^{n-1}(b_j^k)$ is the intrinsic information.

### 2.3.1 Pulse Detector

A pulse detector [10] is the first stage in the iterative turbo multiuser detector. The information symbols corresponding to each pulse interval of the $k^{th}$ user $b_j^k$'s are assumed to be independent. Hence, a priori information $b_j^k = \ldots = b_{N_j}^k$ is ignored in the pulse detector. At the $n^{th}$ iteration, the pulse detector computes the a posteriori LLR of $b_j^k$, given the received signal, the information about the transmitted bits from other users, and the a priori information about $b_j^k$ obtained from the symbol detector as

$$\lambda_1^n(b_j^k) = \log \frac{P(b_j^k = 1 | r)}{P(b_j^k = -1 | r)}$$

$$= \log \frac{P(r | b_j^k = 1)}{P(r | b_j^k = -1)} + \log \frac{P(b_j^k = 1)}{P(b_j^k = -1)}$$

$$= \log \frac{f(r_{(j,k)} | b_j^k = 1)}{f(r_{(j,k)} | b_j^k = -1)} + \log \frac{P(b_j^k = 1)}{P(b_j^k = -1)}$$  \hspace{1cm} (2.3.3)

where $r_{(j,k)}$ is defined in equation (2.2.4) and it is assumed that each pulse is modulated with an independent symbol. The a priori LLR, $\log \frac{f(r_{(j,k)} | b_j^k = 1)}{f(r_{(j,k)} | b_j^k = -1)}$ is defined in (2.3.4) and can be derived from equation (2.2.4), where in equation (2.3.4), $[b]_g$ denotes the $g^{th}$ element of $b$.

$$\log \frac{f(r_{(j,k)} | b_j^k = 1)}{f(r_{(j,k)} | b_j^k = -1)} = \log \frac{\sum_{b^{(\pm1)}} e^{(r_{(j,k)} - 1) b_j^k b_j^k} \prod_{g=1}^{K_j^k} \frac{1 + [b]_g \tanh \left( \frac{1}{2} \lambda_2^{g-1} \left( b_j^k (g) \right) \right)}{\prod_{g=1}^{K_j^k} \frac{1 + [b]_g \tanh \left( \frac{1}{2} \lambda_2^{g-1} \left( b_j^k (g) \right) \right)}}}{\sum_{b^{(\pm1)}} e^{(r_{(j,k)} - 1) b_j^k b_j^k} \prod_{g=1}^{K_j^k} \frac{1 + [b]_g \tanh \left( \frac{1}{2} \lambda_2^{g-1} \left( b_j^k (g) \right) \right)}}$$  \hspace{1cm} (2.3.4)
2.3.2 Symbol Detector

The symbol detector is essentially the same as that discussed by Fishler and Poor [10]. The symbol detector [10] exploits the fact that \( b_i^k = ... = b_{N_f}^k \). It computes the \textit{a posteriori} LLR of \( b_j^k \), given the information from the pulse detector as

\[
\Lambda_2^k(b_j^k) = \log \frac{P(b_j^k = 1 | \lambda_1^u(b_j^k), j = 1, ..., N_f)}{P(b_j^k = -1 | \lambda_1^u(b_j^k), j = 1, ..., N_f)}
\]

\[
= \log \frac{P(b_1^k = ... b_{N_f}^k = 1 | \lambda_1^u(b_j^k), j = 1, ..., N_f)}{P(b_1^k = ... b_{N_f}^k = -1 | \lambda_1^u(b_j^k), j = 1, ..., N_f)}
\]

\[
= \sum_{l=1}^{N_f} \log \frac{P(b_l^k = 1 | \lambda_1^u(b_l^k))}{P(b_l^k = -1 | \lambda_1^u(b_l^k))} + \lambda_1^u(b_j^k)
\]

\[
= \sum_{l=1, l \neq j}^{N_f} \lambda_1^u(b_l^k) + \lambda_1^u(b_j^k)
\]

The output of the symbol detector \( \Lambda_2^k(b_j^k) \) contains the extrinsic information \( \lambda_2^u(b_j^k) \) used by the pulse detector as the \textit{a priori} information and the intrinsic information \( \lambda_1^u(b_j^k) \) from all pulses from same user \( k \) bearing the same information symbol.
Chapter 3

Convolutional Coding and Decoding

3.1 Introduction

This section introduces channel coding and its brief history to the reader who is not familiar with channel coding concepts. The explanation and derivations discussed here follow very closely Hanzo et al. [12].

The history of channel coding, or forward error correction (FEC) coding refers back to the Shannon period. Shannon, although he predicted that a reliable communication could be achieved by FEC, refrained from proposing explicit practical implementation schemes.

3.1.1 Convolutional Encoding

Convolutional codes (CC) can be classified as systematic and non-systematic. In the case of a systematic convolutional encoder, the encoder encodes the information bits/symbol and outputs the coded bits in such a way that part of the coded bits consists of the information bits.

In general, a convolutional encoder will have $K$ shift registers when it has to encode $k$ information bits. The number of $K$ shift registers in an encoder is called constraint length, and the memory of the convolutional code is $K + 1$. If an encoder outputs “$n$” coded bits from “$k$” input bits, then the rate of the encoder is defined by $R = \frac{k}{n}$, implying that $R \leq 1$. In general, CC is denoted as $CC(n, h, k)$, given that $n$, the code, is fully specified. The $n$
generator polynomial is necessary to describe the topology of the modulo 2 gates generating
the output bits of the convolutional encoder. The generator polynomials are binary, where
1 or 0 indicates the presence or absence, respectively, of a specific link as shown in Figure
3.1. The generator polynomials are constituted by $g_1 = [1 \ 0 \ 0]$ and $g_2 = [1 \ 1 \ 1]$, or

\[ g_1(z) = 1 + 0 \cdot z^1 + 0 \cdot z^2 \] \[ g_2(z) = 1 + z^1 + z^2. \]

Figure 3.1: Convolutional encoder CC(2,1,2).

in an equivalent polynomial $g_1(z) = 1 + 0 \cdot z^1 + 0 \cdot z^2$ and $g_2(z) = 1 + z^1 + z^2$.

Figure 3.2 shows the encoder operation for the first ten clock cycles tabulating the input
bits, the shift register $S_1, S_2$, and the corresponding output bits.

<table>
<thead>
<tr>
<th>Input</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 3.2: Input and output bits of CC(2,1,2).

### 3.1.2 State and Trellis Diagrams

The behavior of the convolutional encoder in Figure 3.3 is shown in a state diagram in Figure
3.4. At any given point of time, there are two bits in the two shift register of the encoder.
Hence, there are four possible states in the state diagram, which is governed by an incoming bit $b_i$. The transition from one state to another, due to the input bit “0” is denoted by a solid line, and due to the input bit “1” is denoted by a dotted line. The associated encoded output in bits is also plotted above the line associated with each of the transitions in Figure 3.4.

![Figure 3.3: RSC encoder](image)

**3.2 The Maximum A-Posteriori Algorithm**

The convolutional encoder encodes the information bits to coded bits, which are then transmitted across the noisy channel and decoded by the receiver. This decoding can be based either on a hard decision or a soft decision. In a hard decision, a binary decision is made concerning the received bits and is probably the simplest way to correct these errors. The most popular hard-decision algorithm is the viterbi algorithm (VA). This algorithm tries to compare the received bits with all possible sequence, of the trellis diagram and evaluate the probability of each of the associated paths. Thus, it assigns the probability related quantity to a specific decoded sequence.

The Maximum A-Posteriori (MAP) algorithm is another alternative decoding algorithm. This algorithm was first proposed by Bahl et al. in 1974 [14]. The MAP algorithm outputs
the finely graded multilevel confidence measure concerning the probability of a binary one and a binary zero. Hence, its output is considered a soft-decision output. The performance of this algorithm is much better than that of the viterbi algorithm. The main difference between the viterbi algorithm and the MAP algorithm is that the viterbi algorithm minimizes the number of groups of bits associated with the trellis path, rather than the actual number of bits, which are decoded incorrectly. However, the MAP algorithm provides not only the estimate of bit sequence, but also the probability for each bit that has been decoded correctly. Finally, the soft-decision algorithms derived from the MAP algorithm are the log-MAP algorithm. The log-MAP algorithm is also called the soft-input soft-output (SISO) decoding algorithm.

3.2.1 Mathematics Preliminaries

This section explains the theory behind the MAP algorithm that is used for the soft-output decoding of the component convolutional codes. It is assumed here that binary codes are used. The explanation and examples follow very closely to those of Hanzo et al. [12].

The Bayes rule is repeatedly used in this section. According to this rule, the joint probability of $a$ and $b$, $P(a \land b)$, in terms of the conditional probability of $a$ given $b$ is given as

$$P(a \land b) = P(a|b) \times P(b)$$  \hspace{1cm} (3.2.1)

and

$$P(\{a \land b|c\}) = P(a|\{b \land c\}) \times P(b|c)$$  \hspace{1cm} (3.2.2)

The MAP algorithm gives the probability that each decoded bit $u_k$ was $+1$ or $-1$, given the received symbol sequence $y$. This is exactly the same as finding the a posteriori LLR $L(u_k|y)$, where

$$L(u_k|y) = \log \frac{P(u_k = +1|y)}{P(u_k = -1|y)}$$  \hspace{1cm} (3.2.3)
Using Bayes rule, this equation can be rewritten as

\[ L(u_k|y) = \log \frac{P(u_k = +1 \land y)}{P(u_k = -1 \land y)} \]  

(3.2.4)

Considering the state transition diagram shown in Figure 3.4, the transition associated with input bit \(-1\) is shown as a continuous line and the transition corresponding to input bit of \(+1\) is shown as a broken line.

Figure 3.4: Possible state transition in RSC component code.

The input bit \(u_k\), which results in the transition, can be easily determined if the previous state, \(S_{k-1}\) and current state, \(S_k\) are known. Hence, the probability that \(u_k = +1\) is equal to the probability that the transition from the previous state \(S_{k-1}\) to the present state \(S_k\) is one of the set of four possible transitions that can occur when \(u_k = +1\). This set of four possible transitions is mutually exclusive. Hence, the probability that any one of them occurs is equal to the sum of their individual probabilities. Therefore, equation (3.2.4) can be rewritten as

\[ L(u_k|y) = \log \frac{\sum_{(\hat{s}, s) \Rightarrow u_k = +1} P(S_{k-1} = \hat{s} \land S_k = s \land y)}{\sum_{(\hat{s}, s) \Rightarrow u_k = -1} P(S_{k-1} = \hat{s} \land S_k = s \land y)} \]  

(3.2.5)

where \((\hat{s}, s) \Rightarrow u_k = +1\) is the set of transitions that can occur when input bit \(u_k = +1\) from \(S_{k-1} = \hat{s}\) to \(S_k = s\). The received sequence \(y\) is divided into three sections: the received
sequence associated with the present transition $y_k$, the received sequence prior to the present transition $y_{j<k}$, and the received sequence after the present transition $y_{j>k}$. Figure 3.5 shows the example for the $K = 3$RSC component code shown in figure 3.3.

By denoting the numerator and denominator in equation (3.2.5) with a brief notation $P(\hat{s} \land s \land y)$, then the individual probabilities can be written as,

$$P(\hat{s} \land s \land y) = P(\hat{s} \land s \land y_{j<k} \land y_k \land y_{j>k})$$

(3.2.6)

By assuming the channel as memoryless we can rewrite equation (3.2.6) as

$$P(\hat{s} \land s \land y) = \alpha_{k-1}(\hat{s})$$

(3.2.7)

where

$$\alpha_{k-1}(\hat{s}) = P(S_{k-1} = \hat{s} \land y_{j<k})$$

(3.2.8)
is the probability that the trellis is in state \( s \) at time \( k - 1 \), and the received channel sequence up to this point is \( y_{j<k} \), as shown in Figure 3.5.

\[
\beta_k(s) = P(y_{j>k} | S_k = s) \tag{3.2.9}
\]

is the probability that given the trellis is in the state \( s \) at time \( k \), the future received channel sequence will be \( y_{j>k} \), and

\[
\gamma_k(\hat{s}, s) = P(y_k \land S_k = s | S_{k-1} = \hat{s}) \tag{3.2.10}
\]

is the probability that given the trellis was in the state \( \hat{s} \) at time \( k - 1 \), it moves to state \( s \) and the received channel sequence for this transition is \( y_k \).

### 3.2.2 Calculation of the \( \alpha_k(s) \) Values

The \( \alpha_k(s) \) value from the definition of \( \alpha_{k-1}(s) \) in equation (3.2.8) can be written as.

\[
\alpha_k(s) = P(S_k = s \land y_{j<k+1})
\]

\[
= P(s \land y_{j<k} \land y_k)
\]

\[
= \sum_{\text{all } \hat{s}} P(s \land \hat{s} \land y_{j<k} \land y_k) \tag{3.2.11}
\]

Under the assumption that the channel is memoryless and using the Bayes rule

\[
\alpha_k(s) = \sum_{\text{all } \hat{s}} P(\{y_k \land s\} \mid \{\hat{s} \land y_{j<k}\}) \cdot P(\hat{s} \land y_{j<k})
\]

\[
= \sum_{\text{all } \hat{s}} P(\{y_k \land s\} \mid \hat{s}) \cdot P(\hat{s} \land y_{j<k})
\]

\[
= \sum_{\text{all } \hat{s}} \gamma_k(\hat{s}, s) \cdot \alpha_{k-1}(\hat{s}). \tag{3.2.12}
\]

The \( \alpha_k(s) \) value can be calculated recursively when once \( \gamma_k(\hat{s}, s) \) values are known. Assuming that the trellis has the initial state \( S_0 = 0 \), the initial conditions for this recursion are

\[
\alpha_0(S_0 = 0) = 1 \tag{3.2.13}
\]

\[
\alpha_0(S_0 = s) = 0 \text{ for all } s \neq 0. \tag{3.2.14}
\]
3.2.3 Calculation of the $\beta_k(s)$ Values

The values of $\beta_k(s)$ are calculated recursively as shown here. The $\beta_k(s)$ value from the definition of $\beta_{k-1}(\delta)$ can be rewritten as

$$\beta_{k-1}(\delta) = P(y_{j>k-1}|S_{k-1} = \delta) \quad (3.2.15)$$

and splitting a single probability into the sum of joint probabilities and using the derivation from the Bayes rule in equation (3.2.2) gives.

$$\beta_{k-1}(\delta) = P(y_{j>k-1}|\delta) = \sum_{\text{all } s} P\{y_{j>k-1} \land s\}|\delta)$$

$$= \sum_{\text{all } s} P\{y_k \land y_{j>k} \land s\}|\delta)$$

$$= \sum_{\text{all } s} P\{y_{j>k}|\delta \land s \land y_k\} \cdot P\{y_k \land s\}|\delta)$$

$$= \sum_{\text{all } s} P(y_{j>k}|\delta) \cdot P\{y_k \land s\}|\delta)$$

$$= \sum_{\text{all } s} \beta_k(s) \cdot \gamma_k(\delta, s) \quad (3.2.16)$$

The $\beta_{k-1}(\delta)$ value can be calculated from the $\beta_k(\delta)$ value recursively (backward) when once the $\gamma_k(\delta, s)$ values are known. Unlike $a_0(s)$, the initial condition that should be used for $\beta_N(s)$ is very clear. Breiling [13] points out that if you consider $\beta_{N-1}(s)$ from equation (3.2.9) then

$$\beta_{N-1}(\delta) = P(y_N|\delta)$$

$$= \sum_{\text{all } s} P\{y_N \land s\}|\delta)$$

$$= \sum_{\text{all } s} \gamma_N(\delta, s) \quad (3.2.17)$$

and from the backward recursion for $\beta_{k-1}(\delta)$ in equation (3.2.16).

$$\beta_{N-1}(\delta) = \sum_{\text{all } s} \beta_k(s) \gamma_N(\delta, \delta) \quad (3.2.18)$$

If both the equations (3.2.18) and (3.2.17) are to be satisfied, then

$$\beta_N(\delta) = 1 \quad \text{for all } s \quad (3.2.19)$$
3.2.4 Calculation of the $\gamma_k(\hat{s}, s)$ Values

The derivation for the calculation of $\gamma_k(\hat{s}, s)$ values closely follows to that of Hanzo et al. [12]. By using the definition of Bayes rule and the $\gamma_k(\hat{s}, s)$ value in equation (3.2.10)

$$\gamma_k(\hat{s}, s) = P(\{y_k \land s\} | \hat{s})$$

$$= P(y_k | \{\hat{s} \land s\}) \cdot P(s | \hat{s})$$

$$= P(y_k | \{\hat{s} \land s\}) \cdot P(u_k)$$

(3.2.20)

where input bit $u_k$ results in the transition from state $S_{k-1} = \hat{s}$ to state $S_k = s$, and $P(u_k)$ is the a-priori probability of this bit. From the derivations in the Hanzo et al. [12] and from equations (3.2.6) and (3.2.7) we can write the conditional LLR of $u_k$ given the received sequence as

$$L(u_k | y) = \log \frac{\sum_{(\hat{s}, s) \Rightarrow u_k = +1} P(S_{k-1} = \hat{s} \land S_k = s \land y)}{\sum_{(\hat{s}, s) \Rightarrow u_k = -1} P(S_{k-1} = \hat{s} \land S_k = s \land y)}$$

$$= \log \frac{\sum_{(\hat{s}, s) \Rightarrow u_k = +1} \beta_k(s) \cdot \gamma_k(\hat{s}, s) \cdot \alpha_{k-1}(\hat{s})}{\sum_{(\hat{s}, s) \Rightarrow u_k = -1} \beta_k(s) \cdot \gamma_k(\hat{s}, s) \cdot \alpha_{k-1}(\hat{s})}$$

(3.2.21)

Following this brief introduction to the concepts of convolutional encoding and its mathematics preliminaries is the next chapter on the proposed iterative receiver model.
Proposed Iterative Receiver Model

4.1 Introduction

Chapter 3 derived the turbo-principled multiuser detector which had only two stages. The proposed iterative receiver structure here has three stages, as shown in Figure 4.1. The first two stages are very similar to the turbo-principled multiuser detector: the first stage that begins the pulse detector and the second stage as the symbol detector. Output of symbol decoder is fed to the channel decoder instead of the pulse detector. The channel decoder acts as a third stage and its output is fed to the pulse detector. The soft a priori information exchanges among these three stages.

The pulse detector at the first stage takes the output of the linear-matched filter front-end as its input. It will be shown that at the $n^{th}$ iteration, the pulse detector computes the a posteriori log-likelihood ratio of the pulse $b^k_j$, denoted as $\Lambda^n_1(b^k_j)$, given the received signal, other users' information about the transmitted bits, and the a priori LLR $b^k_j$ obtained from the channel decoder as

$$\Lambda^n_1(b^k_j) = \Lambda^n_1(b^k_j) + \lambda^{n-1}_3(b^k_j).$$

(4.1.1)

where $\Lambda^n_1(b^k_j)$ is the extrinsic information, and $\lambda^{n-1}_3(b^k_j)$ represents the a priori LLR of $b^k_j$, which is computed by the channel detector at the $(n-1)^{th}$ iteration. For the first iteration, $\lambda^0_3(b^k_j) = 0$ for $k = 1, 2, ..., K$ and $j = 1, 2, ..., N_f$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The symbol detector computes the \textit{a posteriori} LLR of $b_j^k$, denoted by $\Lambda_2^k(b_j^k)$, given the soft information $\lambda_1^k(b_j^k)$ provided from the pulse detector as

$$\Lambda_2^k(b_j^k) = \lambda_2^k(b_j^k) + \lambda_1^k(b_j^k)$$ \hspace{1cm} (4.1.2)

where $\lambda_2^k(b_j^k)$ is the extrinsic information that is fed to the channel decoder, and $\lambda_1^k(b_j^k)$ is the intrinsic information received from the pulse detector at the first stage.

Similarly, it will be shown that at the $n^{th}$ iteration, the channel decoder computes the \textit{a posteriori} LLR of $b_j^k$, denoted by $\Lambda_n^k(b_j^k)$, given the soft information $\lambda_2^{n-1}(b_j^k)$ from the symbol detector as

$$\Lambda_n^k(b_j^k) = \lambda_n^k(b_j^k) + \lambda_2^{n-1}(b_j^k)$$ \hspace{1cm} (4.1.3)

where $\lambda_3^k(b_j^k)$ is the extrinsic information, and $\lambda_2^{n-1}(b_j^k)$ is the intrinsic information delivered from the symbol detector.
It should be noted that the notations used in Chapter 3 were offered to provide a brief introduction to the concepts of convolutional coding and decoding. These notations could vary slightly from the notations used in this chapter.

4.2 Proposed Model

4.2.1 Pulse Detector

The first stage of the proposed iterative turbo multiuser receiver is a pulse detector similar to that of Fishler and Poor [10]. The information symbols corresponding to each pulse interval of the $k^{th}$ user $b^k_i$'s are assumed to be independent. Hence, a priori information $b^k_i = ... = b^k_N$ is ignored in the pulse detector. At the $n^{th}$ iteration, the pulse detector computes the a posteriori LLR of $b^k_j$, given the received signal, the information about the transmitted bits from other users, and the a priori information about $b^k_j$ obtained from the channel decoder as

$$
\Lambda^n_i(b^k_j) = \log \frac{P(b^k_j = 1 | r)}{P(b^k_j = -1 | r)}
$$

$$
= \log \frac{P(r | b^k_j = 1)}{P(\mathbf{r} | b^k_j = -1)} + \log \frac{P(b^k_j = 1)}{P(b^k_j = -1)}
$$

$$
= \log \frac{p(r_{(j,k)} | b^k_j = 1)}{p(r_{(j,k)} | b^k_j = -1)} + \log \frac{P(b^k_j = 1)}{P(b^k_j = -1)}
$$

$$
= \lambda^n_i(b^k_j) + \lambda^{n-1}_i(b^k_j),
$$

(4.2.1)

where $r_{(j,k)}$ is defined in equation (2.2.4), and it is assumed that each pulse is modulated with an independent symbol; the a priori LLR, $\log \frac{p(r_{(j,k)} | b^k_j = 1)}{p(r_{(j,k)} | b^k_j = -1)}$, is defined in equation (4.2.2) and can be derived from equation (2.2.4). In equation (4.2.2), $[b]_g$ denotes the $g^{th}$ element of $b$.

$$
\log \frac{f(r_{(j,k)} | b^k_j = 1)}{f(r_{(j,k)} | b^k_j = -1)} = \log \frac{\sum_{b^k_{-1} b^k_{-2} ... b^k_{-M}} e^{\frac{1}{2} r_{(j,k)}^T \mathbf{H}^{-1} r_{(j,k)} + \mathbf{r}_{(j,k)}^T \mathbf{b}}}{\sum_{b^k_{-1} b^k_{-2} ... b^k_{-M}} e^{\frac{1}{2} r_{(j,k)}^T \mathbf{H}^{-1} r_{(j,k)} + \mathbf{r}_{(j,k)}^T \mathbf{b}}}
$$

$$
= \log \frac{\prod_{g=1}^{K_j} \tanh \left( \frac{1}{2} \lambda_{g-1} \left( b^k_j + [b]_g \right) \right)}{\prod_{g=1}^{K_j} \tanh \left( \frac{1}{2} \lambda_{g-1} \left( b^k_j - [b]_g \right) \right)}
$$

(4.2.2)
4.2.2 Symbol Detector

The symbol detector is essentially the same as that discussed in Fishler and Poor [10]. The symbol detector of Fishler and Poor [10] exploits the fact that \( b^k_1 = \ldots = b^k_{N_f} \). It computes the \textit{a posteriori} LLR of \( b^k_j \), given the information from the pulse detector as

\[
\Lambda_2^v(b^k_j) = \log \frac{P(b^k_j = 1|\lambda^u_1(b^k_j), j = 1, \ldots, N_f)}{P(b^k_j = -1|\lambda^u_1(b^k_j), j = 1, \ldots, N_f)}
\]

\[
= \log \frac{P(b^k_j = b^k_{N_f} = 1|\lambda^u_1(b^k_j), j = 1, \ldots, N_f)}{P(b^k_j = -1|\lambda^u_1(b^k_j), j = 1, \ldots, N_f)}
\]

\[
= \sum_{i=1}^{N_f} \log \frac{P(b^k_j = 1|\lambda^u_1(b^k_j))}{P(b^k_j = -1|\lambda^u_1(b^k_j))} + \lambda^v_2(b^k_j)
\]

\[
= \sum_{i=1, \neq j}^{N_f} \lambda^u_1(b^k_j) + \lambda^v_2(b^k_j). \quad (4.2.4)
\]

The output of the symbol detector \( \Lambda_2^v(b^k_j) \) contains the extrinsic information \( \lambda^v_2(b^k_j) \) used by the channel decoder as the \textit{a priori} information (after deinterleaving) and the intrinsic information \( \lambda^u_1(b^k_j) \) from all pulses from the same user \( k \) bearing the same information symbol.

4.2.3 Channel Decoder

The third stage is the channel decoder and has a bank of \( K \) decoders, as shown in Figure 4.1. A code bit interleaver and deinterleaver are used to reduce the influence of error bursts at the output and input of each channel decoder. The input to each of these decoders is the deinterleaved \textit{a priori} output from the symbol detector. The channel decoder computes the LLR of the coded bits and the information bits. The convolutional encoder with a binary rate of \( \frac{k_o}{n_o} \) at the transmitter end encodes a block of \( k_o \) information bits and outputs \( n_o \) coded bits. At time \( t \), if the input to the encoder is \( d^t = [d^t_1, \ldots, d^t_{k_o}] \), then the output is
Denote $\mathbf{d}(s', s)$ as the input bits which results in the transition of the encoder trellis state $S_{t-1} = s'$ to $S_t = s$ and outputs the $n_o$ coded bits $\mathbf{b}(s', s)$. Letting $\tau$ be the code block length (after padding zeros) and the output of the channel encoder at time $t$ denoted by $\mathbf{b}_t$ then the notation by Wang and Poor [11]

$$P[\mathbf{b}_t | s', s] \triangleq P[\mathbf{b}_t = \mathbf{b}(s', s)]$$

(4.2.5)

can be used to define the forward and backward recursions as [11], [14]

$$\alpha_t(s) = \sum_{s'} \alpha_{t-1}(s') P[\mathbf{b}_t | s', s] \quad t = 1, 2, \ldots, \tau$$

(4.2.6)

$$\beta_t(s) = \sum_{s'} \beta_{t+1}(s') P[\mathbf{b}_{t+1} | s', s] \quad t = \tau - 1, \tau - 2, \ldots, 0$$

(4.2.7)

with boundary conditions $\alpha_0(s) = 1$, $\alpha_t(s) = 0$, and $\beta_\tau(s) = 1$, $\beta_t(s) = 0$. To obtain a numerically stable algorithm, the parameters $\alpha_t(s)$ and $\beta_t(s)$ can be scaled as the computation proceeds as explained by Wang and Poor [11]. That is, $\tilde{\alpha}_t(s)$, which is the scaled version of $\alpha_t(s)$ is computed as follows: first set $\tilde{\alpha}_1(s) = \alpha_1(s)$ and $\tilde{\alpha}_t(s) = c_t \tilde{\alpha}_t(s)$ with $c_t \triangleq \frac{1}{\sum s', \tilde{\alpha}_t(s')}$. Then, $\tilde{\alpha}_t(s)$ is computed using

$$\tilde{\alpha}_t(s) = \sum_{s'} \tilde{\alpha}_{t-1}(s') P[\mathbf{b}_t | s', s']$$

(4.2.8)

Similarly,

$$\tilde{\beta}_t(s) = \sum_{s'} \tilde{\beta}_{t+1}(s') P[\mathbf{b}_{t+1} | s', s']$$

(4.2.9)

$$\hat{\beta}_t(s) = c_t \tilde{\beta}_t(s)$$

(4.2.10)

The SISO channel decoder of the $k^{th}$ user outputs the a posteriori LLR of the code bit $b^q_k$, for $q = 1, 2, \ldots, n_o$ computed as (note that the user index $k$ has been suppressed)

$$\Lambda_q[b^q_k] \triangleq \log \frac{\sum_{s'} \alpha_{t-1}(s') \beta_t(s) \prod_{i=1}^{n_o} P[b_i^q | s', s]}{\sum_{s'} \alpha_{t-1}(s') \beta_t(s) \prod_{i=1}^{n_o} P[b_i^q | s', s]}$$

$$= \log \left( \frac{\sum_{s'} \alpha_{t-1}(s') \beta_t(s) \prod_{i \neq q} P[b_i^q | s', s]}{\sum_{s'} \alpha_{t-1}(s') \beta_t(s) \prod_{i \neq q} P[b_i^q | s', s]} \right) + \log \frac{P[b^q_k = +1]}{P[b^q_k = -1]}$$

(4.2.11)
where $S_q^+$ is defined as the set of state pairs $(s', s)$ such that the $q^{th}$ bit of the code symbol $b(s', s)$ is $+1$ and similarly for $S_q^-$. Hence, we can write $\Lambda_3[b_q^+]$ as

$$\Lambda_3[b_q^+] = \lambda_3[b_q^+] + \lambda_2^*[b_q^-]$$

where $\lambda_3[b_q^+]$ is the extrinsic information of the SISO channel decoder, and $\lambda_2^*[b_q^-]$ is the a priori information provided by the symbol detector.

The LLR of information bits is needed as the final output of the channel decoder in the last iteration. This can be obtained by modifying equation (4.2.11): instead of summing over $S_q^+$ and $S_q^-$, the summation is performed over $U_q^+$, the set of state pairs $(s', s)$, such that the $q^{th}$ bit of the information symbol $d(s', s)$ is $+1$ and similarly for $U_q^-$. Then,

$$\Lambda_3[d_q^+] \triangleq \log \frac{\sum_{U_q^+} \alpha_{q-1}(s') \beta_q(s) \prod_{i=1}^{n_q} P[b_i(s', s)]}{\sum_{U_q^-} \alpha_{q-1}(s') \beta_q(s) \prod_{i=1}^{n_q} P[b_i(s', s)]} + \log \frac{P[b_q^+ = +1]}{P[b_q^+ = -1]}.$$

Note that the input to the pulse detector is the soft values of the pulses, not the symbols. However, the channel decoder outputs the LLR of coded bits $\Lambda_3[b_q^+]$. Since it is assumed that the pulses are independent in the pulse detector, the channel decoder output is divided by $N_f$ and fed them back to the pulse detector as the a priori information of the $N_f$ pulses $\lambda_3[b_f^+]$ (after interleaving). The iteration then continues through these three stages.
Chapter 5

Performance Analysis

This section presents the simulation results of the proposed iterative multiuser detector. The average bit error rate of the proposed iterative receiver is evaluated as a function of signal-to-noise ratio (SNR) per pulse, and the number of users in the system. The UWB system considered has $N_c = 10$ and $N_f = \{10, 20\}$ and the UWB pulses are generated as discussed by Benedetto and Giancola [1]. All user information bits are encoded with a rate $\frac{1}{2}$ convolutional encoder having constraint length of $\nu = 5$ and the generator matrix [23, 35] in octal notation. The information block size is 128 bits. The same random interleaver is used for each user. All users are assumed to have equal received power.

Figures 5.1 and 5.2 show the average bit error rates of the proposed iterative multiuser receiver as a function of SNR per pulse with $K = 20$ and $K = 10$ users, respectively. In figure 5.1, the number of users in the system is $K = 20$ with $N_c = 20$ and $N_f = 10$. Performance has increased greatly after only two iterations, but there is still a performance degradation compared to a single-user system, due to residual multiple-access interference.

In Figure 5.2, the number of users in the system is $K = 10$ with $N_c = 20$ and $N_f = 10$. It can be seen that the proposed iterative receiver system performs close to the single-user system after only two iterations. The performance shown in Figure 5.2 is comparatively better than the performance shown in Figure 5.1 as the number of users considered in Figure 5.2 is less than that in Figure 5.1.
Figure 5.1: Average probability of turbo UWB receiver with $N_c = 20$, $N_f = 10$, and $K = 20$.

Figure 5.2: Average probability of turbo UWB receiver with $N_c = 20$, $N_f = 10$, and $K = 10$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 5.3: Average probability of turbo UWB receiver with $N_c = 10$, $N_f = 10$, and $K = 10$.

Figure 5.3 demonstrates the average bit error rates of the iterative multiuser receiver as a function of SNR per pulse with $N_f = 10$, $N_c = 10$, for $K = 10$ users. Since $N_c$, the number of chips per pulse, is less than those considered in Figure 5.2, there is more interference in this system. As a result, in the first iteration, the performance of this system has considerably degraded compared to the system considered in Figure 5.2. However, the performance of the proposed receiver quickly improves with more iterations, although the final performance is not as good as that seen in Figure 5.2.
Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis proposed a iterative receiver structure for decoding multiuser information data in a convolutionally coded UWB system. At each iteration, extrinsic information was extracted from a pulse detector, symbol detector, and channel decoder and then used as a priori information in the next iteration, similar to turbo decoding. The low-complexity multiuser detector cancels interference among the different users. The performance of this proposed receiver structure is demonstrated in the simulation results. It can be seen that the proposed iterative receiver structure offers a performance that is very close to that of the single-user bound at a high SNR.

6.2 Future work

A receiver structure was proposed where the effects of multipath and fading on signal amplitudes are not considered. In general, UWB signals in indoor communications undergo log-normal fading and travel in more than one path. This would be a straightforward extension of the proposed scheme to work in a multipath and fading environment.

This scheme could also be analyzed with concepts derived from factor graphs and sum-product algorithms. In addition, convergence analysis could be done through exit charts.
REFERENCES
REFERENCES


