LOW-COMPLEXITY ITERATIVE RECEIVERS FOR MULTIUSER SPACE-TIME BLOCK CODING SYSTEMS

A Thesis Submitted
to the College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Electrical and Computer Engineering
University of Saskatchewan

by

Yajun Yang

Saskatoon, Saskatchewan, Canada

Permission To Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, it is agreed that the Libraries of this University may make it freely available for inspection. Permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professors who supervised this thesis work or, in their absence, by the Head of the Department of Electrical and Computer Engineering or the Dean of the College of Graduate Studies and Research at the University of Saskatchewan. Any copying, publication, or use of this thesis, or parts thereof, for financial gain without the written permission of the author is strictly prohibited. Proper recognition shall be given to the author and to the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

Request for permission to copy or to make any other use of material in this thesis in whole or in part should be addressed to:

Head of the Department of Electrical and Computer Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 5A9

Acknowledgments

The first person I would like to thank is my direct supervisor, Prof. Ha H. Nguyen. His overly enthusiastic and integral view on research and his mission for providing "only high-quality work and not less", has made a deep impression on me. The useful discussions and advice from him during our weekly meetings made my research flow easily and smoothly.

I would also like to thank my co-supervisor, Prof. E. Shwedyk, who kept an eye on the progress of my work and always was available when I needed his advise and help. The financial support from his research grant also made this thesis possible.

I am grateful to Prof. E. J. Salt, Prof. K. Takaya, and Prof. B. L. F. Daku. I have learned a lot from their wonderful teaching. I also want to thank all my lab-mates and all of my friends here for their help and encouragement.

I feel a deep sense of gratitude for my wife Honglin Zhang, for her love and patience. It is impossible to finish this thesis without her help and support to take care of our daughter Qinmeng Yang and our son Qinjia Yang, who bring real fun and happiness to our lives. I also want to thank my parents-in-law for their great help to take care of our daughter and son.

Especially, I would like to give my special thanks to my parents, who taught me the good things that really matter in life. Their love and encouragement provide a persistent inspiration for my journey in this life. To them, I dedicate this thesis.

Finally, I wish to express my appreciation to all of those who gave me help when I studied at the University of Saskatchewan.

Abstract

Iterative processing has been shown to be very effective in multiuser space-time block coding (STBC) systems. The complexity and efficiency of an iterative receiver depend heavily on how the log-likelihood ratios (LLRs) of the coded bits are computed and exchanged at the receiver among its three major components, namely the multiuser detector, the maximum *a posterior* probability (MAP) demodulators and the MAP channel decoders. This thesis first presents a method to quantitatively measure the system complexities with floating-point operations (FLOPS) and a technique to evaluate the iterative receiver's convergence property based on mutual information and extrinsic information transfer (EXIT) charts.

Then, an integrated iterative receiver is developed by applying the sigma mappings for M-ary quadrature amplitude modulation (M-QAM) constellations. Due to the linear relationship between the coded bits and the transmitted channel symbol, the multiuser detector can work on the bit-level and hence improves the convergence property of the iterative receiver. It is shown that the integrated iterative receiver is an attractive candidate to replace the conventional receiver when a few receive antennas and a high-order M-QAM constellation are employed.

Finally, a more general two-loop iterative receiver is proposed by introducing an inner iteration loop between the MAP demodulators and the MAP convolutional decoders besides the outer iteration loop that involves the multiuser detection (MUD) as in the conventional iterative receiver. The proposed two-loop iterative receiver greatly improves the iteration efficiency. It is demonstrated that the proposed two-loop iterative receiver can achieve the same asymptotic performance as that of the conventional iterative receiver, but with much less outer-loop iterations.

Table of Contents

List of Figures

Abbreviations

ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BICM	Bit-Interleaved Coded Modulation
BICM-ID	Bit-Interleaved Coded Modulation with Iterative Decoding
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CDMA	Code Division Multiple Access
CSI	Channel State Information
dB	Decibel
DEMO	Demodulator
EXIT Chart	Extrinsic Information Transfer Chart
EF	Error-free Feedback
FLOP	Floating Point Operation
GSM	Global System for Mobile Communications
MAI	Multiple Access Interference
MAP	Maximum A Posteriori Probability
MIMO	Multi-Input Multi-Output
ML	Maximum Likelihood
MLC	Multilevel Coding
MMSE	Minimum Mean Square Error
M-QAM	M-ary Quadrature Amplitude Modulation
M-PSK	M-ary Phase Shift Keying
MUL	Multiplication/Division Operation
MUD	Multiuser Detection
LDPC	Low-Density Parity-Check Code
LLR	Log Likelihood Ratio
PDA	Personal Digital Assistant

PDF	Probability Density Functions
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
SSP	Semi-Set Partitioning
STC	Space-Time Code
STTC	Space-Time Trellis Code
STBC	Space-Time Block Code
TCM	Trellis-Coded Modulation
WCDMA	Wideband Code Division Multiple Access

Notations

$\approx$	approximately equal to
$\triangleq$	defined as
$\cdot$	multiplication operator
x^*	complex conjugate of x
A^{-1}	inverse of matrix A
A^H	Hermitian (conjugate transpose) of matrix A
A^T	transpose of matrix A
$\|x\|$	norm of complex number x
$A_{N\times M}$	a matrix A with N rows and M columns
$\lambda(b)$	*a priori* log-likelihood ratio of bit b
$\Lambda(b)$	an *extrinsic* log-likelihood ratio of bit b
$E\{\cdot\}$	expectation operator
$\log(x)$	natural log of x
$\log_x(y)$	log with base x of y
$\mathcal{C}^N$	the complex N-dimensional space
$\min_x f(x)$	the minimum value of $f(x)$ over all x
$p(X)$	probability density function of random variable X
$P(X=x)$	probability of event $X=x$
$\text{cov}(X,Y)$	covariance of random variables X and Y
$\text{diag}(x_1,\cdots,x_N)$	the $N\times N$ matrix with diagonal elements $x_1,\cdots,x_N$
$I(X,Y)$	mutual information between random variables X and Y
$\{x_j, 1\le j\le J\}$	the vector $X=\{x_1,\cdots,x_J\}$
$\mathcal{CN}(0,1)$	a circularly symmetric complex Gaussian random variable with unit variance
$\text{Real}(x)$	the real part of complex number x
$\text{Imag}(x)$	the imaginary part of complex number x
$\text{Real}(A)$	A matrix consisting of real parts of the elements in matrix A
$\text{Imag}(A)$	A matrix consisting of imaginary parts of elements in matrix A

Γ_{XXX}	the complexity of an iterative receiver named XXX

1. Introduction

Wireless communications first appeared in 1897 with Marconi's successful demonstrations of wireless telegraphy. In the hundred years after that, wireless communications has experienced remarkable evolution with a rapid progress in technology. At the time of this writing, the cellular phone systems, one of the major wireless applications in our lives, are being upgraded to their third generations (CDMA2000, WCDMA) from their second generations (GSM, CDMA95). This upgrade is currently being deployed worldwide to accommodate the rapid growth in both voice traffic and data service. With more and more applications or services provided with diverse wireless facilities, nowadays, we are surrounded by all kinds of wireless devices and networks: cellular phone, hand-held PDA, wireless internet, walkie-talkie, etc. The ultimate goal of wireless communications is to communicate with anybody from anywhere at anytime [1].

While the demand for wireless services is growing at a rapid pace, the available radio bandwidth for wireless applications is extremely limited. It restricts to a great extent the capabilities to increase system capacity, especially for power and complexity limited systems. Bandwidth efficiency is therefore one of the primary concerns in the design of future wireless communications systems.

Multilevel modulation schemes, such as M-ary quadrature amplitude modulation (M-QAM), can increase the spectral efficiency by sending multiple bits per symbol [2]. Unfortunately, the signal transmitted over the wireless channel is subject to severe distortion due mainly to multipath fading. In general, fading refers to the destructive combination of randomly delayed, reflected, scattered, and diffracted signal

components at the receiver [3]. This happens because the transmitted signals travel in different paths from the transmitter to the receiver. Multipath fading leads to serious BER performance degradation for a given modulation technique and makes the achievable capacity of a wireless channel very low. In an additive white Gaussian noise (AWGN) channel, the probability of mistaking a transmitted signal with another one can be made to decrease exponentially with the signal-to-noise ratio (SNR). In contrast, due to the fading effect, the average error probability for a single-antenna wireless system only decreases linearly with the SNR [1]. Therefore, fading compensation is typically required to improve the performance for wireless systems by mitigating the fading effect.

The key feature of any fading compensation technique is to ensure that the receiver be provided with multiple independent received signals that carry the same information [4]. Although some transmitted information may traverse a difficult physical path with deep fading, redundant copies of the information increase the chance that some of the received signals are still good enough to allow reliable detection. This technique is generally called diversity, which exists in different forms, including time diversity, frequency diversity, and space diversity [5].

Time diversity: Here multiple versions of the same signal are received over different time slots. It can be obtained via coding and interleaving. Information is coded and the coded symbols are dispersed over time in different periods so that different parts of the codewords experience independent fades. It therefore provides redundancy in the time domain.

Frequency diversity: In this form of diversity, multiple versions of the same signal are received over different carrier frequencies. Frequency diversity provides redundancy in the frequency domain if the channel is frequency selective.

Antenna diversity: This diversity technique provides redundancy in the spatial domain. It can be achieved when multiple transmit or receive antennas are used and spaced sufficiently far apart. Here, multiple copies of the same information

can be received over different transmit/receive antenna pairs.

After receiving multiple versions of the same transmitted signal, a combining technique is applied to combine all the signal copies in an optimal way to extract as much of the useful information of the transmitted signal as possible before further signal processing takes place [2]. Since diversity is such an important resource, different types of diversities are usually combined to further improve the wireless system performance.

To make a more efficient use of the limited bandwidth, intensive research activities in wireless communications have been carried out and have achieved remarkable progress. Theoretical studies have shown that a much higher average spectral efficiency of wireless transmissions can be reached by employing multiple transmit and/or receive antennas in conjunction with space-time codes (STC) [6] [7] [8]. In fact, this is achieved by taking advantage of the time varying nature of the wireless channel, which was typically considered as a disadvantage in single-antenna systems. This is one of the most significant technical breakthroughs in modern communications. Space-time codes may be split into two main types. One type includes space-time trellis codes (STTCs) [7], which distribute a trellis code over multiple antennas and multiple time-slots and provide both coding gain and diversity gain. The other type includes space-time block codes (STBCs) [8], which act on a block of data at once. STBCs provide only diversity gain, but are much less complex in terms of implementation than STTCs. This thesis only focuses on space-time block codes due to their lower decoding complexity at the receiver.

Recently, multiuser space-time block coding systems have been proposed for wireless systems, which use multiple antennas at the transmitter and the receiver [9]. Space-time block coding is employed together with convolutional coding for data transmission. The key benefit of the scheme with multiple antennas is its ability to turn multiple-path propagation from a disadvantage factor into an advantage factor in wireless communications. With this new scheme, time diversity and antenna

diversity are achieved. Time diversity is achieved by using the convolutional code and a random interleaver. Antenna diversity is achieved by transmitting the signals with multiple transmit antennas and receiving them with multiple receive antennas. Both of these diversity techniques help to combat the fading when the signals are transmitted over the wireless channel.

However the problem is that sharing the limited radio spectrum by employing a multiple access technique makes the system performance degrade to a great extent due to the multiple access interference (MAI). MAI arises because all users simultaneously share the same transmission bandwidth and signals from different users are superimposed over the air. For the multiuser STBC systems considered in [9], no spreading is required (i.e., no bandwidth expansion), and the transmitted signal from one antenna of a user is highly correlated to the signals from other antennas/users. The system performance therefore greatly depends on how well one can differentiate each user's signal and correctly demodulate the information bits for each user based on the received signal and other constraints. In other words, the system performance is determined by how well the multiuser detector, the demodulator and the decoder work and interact. Besides the requirement of good performance, to implement a new technique in practical applications, it is also desired that the technique has a low-complexity, which means that it should be able to detect and decode the information bits with simple processing algorithms.

A significant amount of research work has been carried out in the areas of multiuser detection (MUD) and MAI cancellation. The optimal multiuser detection is the best candidate in terms of the system performance, however its exponential computational complexity with respect to the number of users makes it impractical [10]. Therefore the study of suboptimal multiuser detectors has been very active and quite a few low-complexity multiuser detectors have been proposed.

In particular, an important contribution was presented in [11] to address the MAI problem in coded CDMA systems, where each user employs channel coding (such as convolutional codes). The receiver proposed in [11] combines efficiently the

soft-output MUD and the individual users' soft-input soft-output (SISO) channel decoders in an iterative manner. More recently, this technique was further applied to a multiuser STBC wireless communication system with an M-ary phase-shift keying (M-PSK) constellation [9] and shown to achieve attractive results. The complexity of the iterative receiver is still a main concern to bring the multiuser STBC systems into the practical arena of wireless communications. This challenge is the motivation for our investigation and development of more efficient joint demodulation and decoding techniques for this bandwidth-efficient multiuser system.

1.1 Thesis Contributions

In the first part of the thesis, multiuser STBC systems are introduced and the conventional iterative decoding scheme is described. The algorithm for the conventional receiver [9], which is only given for M-PSK is also extended to a more general case with M-QAM constellation in which the symbols of the constellation have different energies.

In the second part of the thesis, two basic parameters, namely complexity and efficiency, of the investigated multiuser STBC systems are examined. In order to determine the system complexity and compare different detection and demodulation schemes, the complexity is measured in terms of floating point operation (FLOP) following the strategy proposed in [12]. As for the efficiency of the iterative receiver employed by multiuser STBC systems, the extrinsic information transfer characteristic chart (EXIT chart) technique is used to investigate the convergence behavior of the iterative processing. To this end, the concept of mutual information is described first. We then show how to construct the EXIT charts to describe the flow of the extrinsic information among the three modules, namely, the soft-output MUD, the MAP demodulator and the MAP channel decoder. The visualization of the decoding trajectory makes it much easier to compare different iterative receivers, especially when they have the same asymptotic performance and when the error-bound technique cannot work well. Also proposed in this part is a convenient method to calculate

the mutual information between the coded bits and their extrinsic log likelihood ratios (LLR). The proposed method greatly simplifies the calculation by using the estimated histograms of the random variables.

The third part of the thesis is concerned with different iterative receivers for multiuser STBC systems by employing different multiuser detection, demodualtion, and decoding schemes. Three iterative receivers are proposed and discussed to improve the convergence property of the conventional receiver. The first two receivers are related to sigma mapping. One of them is named as the integrated iterative receiver. It exploits the linear relationship of sigma mapping in such a way that the minimum mean-square error (MMSE) demodulator and the MMSE-MUD can be combined into a single module. This allows bit-level multiuser interference cancellation and helps to improve the convergence property of the iterative processing. The third iterative receiver is called the two-loop iterative receiver. By introducing the inner loop into the existing outer loop, it also greatly improves the convergence property of the whole receiver. Thus the iteration times can be reduced in order to approach the same asymptotic error performance. In other words, the system complexity can be greatly reduced compared to the conventional receiver.

1.2 Thesis Organization

The remainder of the thesis is organized as follows.

Chapter 2 introduces the basic multiuser STBC systems including the space-time block codes, the transmitter and receiver's structures and the channel model. An MMSE-MUD is developed for systems with M-QAM constellation. In addition, the MAP demodulator and the MAP decoder for convolutional codes are also presented in this chapter.

Chapter 3 includes two main parts. The first part examines the complexity of three major components (soft-output MUD, MAP demodulator and the MAP channel decoder) employed by the iterative receiver. The second part discusses the efficiency

of the receiver. In this part, the concept of mutual information is first introduced. Then a convenient method to compute the mutual information between the coded bits and their corresponding extrinsic information is presented. Finally, the use of EXIT charts to investigate the efficiency of iterative receivers is described. This chapter provides a good foundation for the following two chapters.

Chapter 4 first presents the sigma mapping and its property. Two different iterative receivers are proposed to exploit the linear relationship between the coded bits and the transmitted channel symbol. The complexity and the corresponding efficiency in terms of convergence behavior of both iterative receivers are examined and compared to the conventional one based on the techniques described in Chapter 3.

Chapter 5 presents another approach to improve the efficiency of the conventional receiver by introducing the two-loop iterative receiver for multiuser STBC systems. The advantage of the two-loop iterative receiver is illustrated by comparing the error performance and the extrinsic information transfer characteristic with those of the conventional receiver.

Finally, Chapter 6 draws the conclusions and gives suggestions for further studies.

2. Multi-User Space-Time Block Coding (STBC) Systems

2.1 System Model

Fig. 2.1 provides an overview of a multiuser STBC system. There are K users in the system, where each user employs N_T transmit antennas. The receiver is equipped with M_R antennas. All users share the same bandwidth. For an efficient use of the limited bandwidth, each user employs an M-QAM constellation and requires no spreading. The details of each user's transmitter are shown in Fig. 2.2. Here $\{b_k(i)\}$, $\{d_k(j)\}$ and $\{c_k(l)\}$ denote the information bit stream, the convolutionally coded bit stream and the modulated symbol stream of the kth user, respectively. The information bits $\{b_k(i)\}$ are first encoded by a convolutional encoder. Then the coded bits $\{d_k(j)\}$ are reordered by a random interleaver. The interleaver is used to eliminate the temporal constraint among the coded bits, which helps to combat burst errors. The interleaved bits are then mapped and modulated to complex symbols $\{c_k(l)\}$ of a general M-QAM constellation. Here $c_k(l) \in \Omega_C \triangleq \{C_1, C_2, \ldots, C_M\}$, where Ω_C denotes a general 2-dimensional M-QAM constellation and $C_k\,(k = 1, \cdots, K)$ denotes one of the signal points of the constellation. The symbol stream $\{c_k(l)\}$ is then partitioned into blocks of symbols and fed to the space-time block encoder to transmit over the channel. Each block contains N symbols. The number of symbols N in one block equals the number of transmit antenna N_T. Therefore, in the rest of this thesis, N is also used to denote the number of transmit antennas.

Space-time blocking coding is a very promising technique, which was developed recently to combat signal fading in wireless channels. The basic idea of space-time

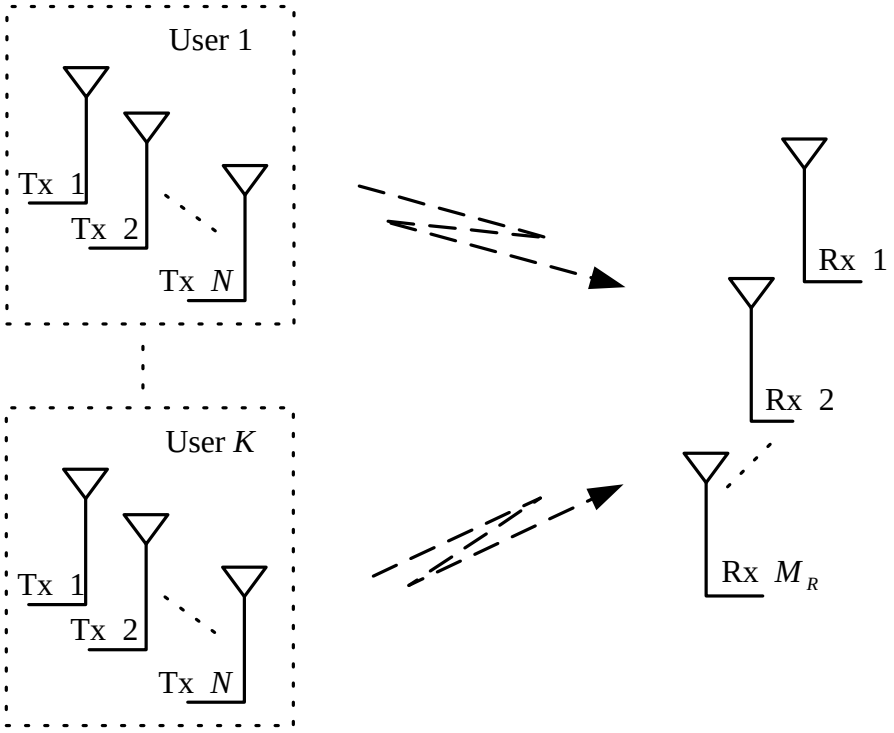


Figure 2.1 Overview of a multiuser STBC system.

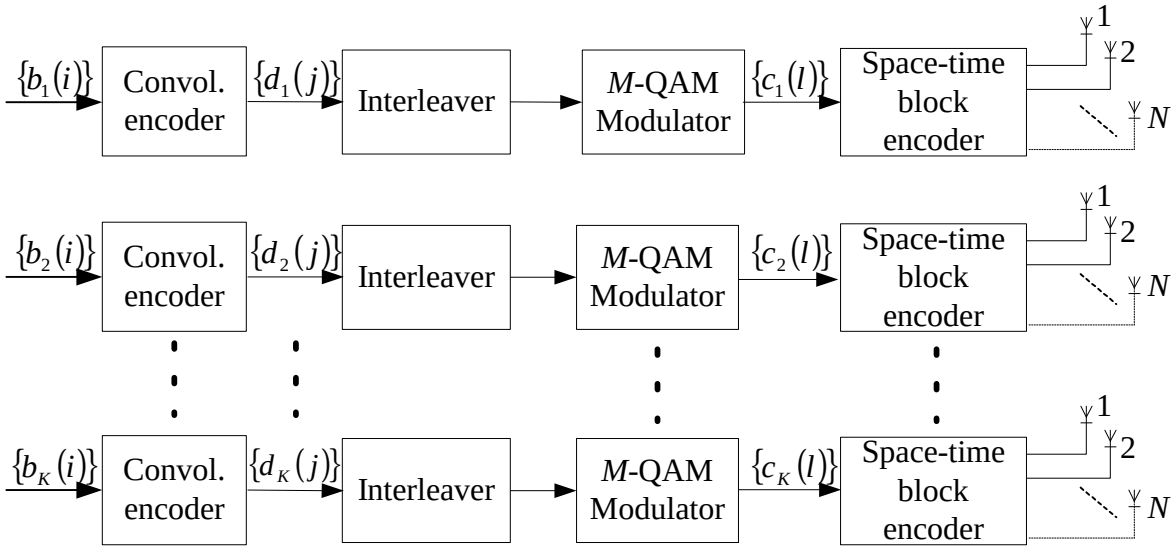


Figure 2.2 Transmitter structure for a multiuser STBC system.

coding is to transmit a vector of code symbols simultaneously from multiple antennas in such a way that independent transmission paths are effectively created. The details of space-time coding will be described in Sections 2.1.2.

2.1.1 Channel Model

As in [9], a quasi-static flat Rayleigh fading channel is assumed for every path from one transmit antenna to one receive antenna. Rayleigh fading channels are a typical model, which apply to wireless radio channels without a line-of-sight path between the transmitter and receiver antennas. Specifically, let $\alpha_{m,n}$ denote the complex fading gain from the nth transmitter antenna to the mth receive antenna, which is modeled as a zero-mean circularly symmetric complex Gaussian random variable with unit variance, i.e.,

$$\alpha_{m,n} \triangleq \mathcal{CN}(0,1),\ n = 1, 2, \cdots, N;\ m = 1, 2, \cdots, M_R \tag{2.1}$$

The quasi-static property means that the channel gain remains constant over one signal block and it varies independently from block to block. Due to the existence of the random interleaver, the symbols in one block can be assumed to be independent and symbol streams of all the users can also be assumed to be mutually independent.

Besides the damaging effect of channel fading, the received signal is also corrupted by noise at the receiver, which can also be modelled as a circularly symmetric complex Gaussian random variable with variance N_0. For one transmitted symbol, the discrete-time baseband received signal can be represented as:

$$y(i) = \alpha(i) \cdot x(i) + n(i) \tag{2.2}$$

where $x(i)$ is the transmitted signal at time slot i, $\alpha(i)$ is the sample of the channel fading gain, $n(i)$ is the sample of the noise at the receiver, and $y(i)$ is the corresponding received signal. Note that when $\alpha(i) = 1$ for all i, the fading channel simplifies to an additive white Gaussian noise (AWGN) channel. For convenience in the following discussion, we also define SNR as the ratio between the average energy of the transmitted symbol $x(i)$ and the noise variance N_0.

2.1.2 Space-Time Block Codes

As mentioned before, space-time block coding is an effective technique used in wireless communications to combat signal degradation caused by channel fading. In

space-time coding, multiple copies of the information data are transmitted across a number of antennas during one period (i.e., at the same time). After passing different paths, all copies of the signal arrive at the receive antenna, and they experience different degrees of degradation. By exploiting the multiple received versions of the transmitted data at the receiver, the transmitted information can be detected with a higher reliability, compared to the case that only one pair of transmit and receive antennas is used in the system. In fact, space-time coding techniques try to combine all the copies of the received signal in an optimal way to extract as much information about the transmitted signal as possible [6].

A space-time block code is defined by a code matrix $\mathbf{G}$. Specifically, to transmit a symbol vector $\underline{c} \triangleq [c(1), c(2), \cdots, c(N)]^T$, the encoder uses the following matrix:

$$\mathbf{G} = \begin{bmatrix} g(1,1) & g(1,2) & \cdots & g(1,N) \\ g(1,1) & g(1,2) & \cdots & g(1,N) \\ \cdots & \cdots & \cdots & \cdots \\ g(P,1) & g(P,2) & \cdots & g(P,N) \end{bmatrix}_{P \times N} \tag{2.3}$$

where P denotes the number of time slots used to transmit one symbol vector $\underline{c}$, N denotes the number of transmit antennas as mentioned before. Each row of matrix $\mathbf{G}$ is a permutation and/or conjugation of the transmitted code vector $\underline{c}$. For transmission, the lth row of $\mathbf{G}$ is transmitted over N antennas during the lth time slot. Note that the code rate of the STBC is N/P symbol per slot.

To clearly illustrate the principle of space-time block coding, consider a simple STBC system with only one user (i.e., K=1). Furthermore, consider $P = N = 2$ and the following famous 2×2 Alamouti code [6]

$$\mathbf{G} = \begin{bmatrix} c(1) & c(2) \\ -c^*(2) & c^*(1) \end{bmatrix} \tag{2.4}$$

where $(\cdot)^*$ denotes the complex conjugate operation. The two channel symbols $c(1)$ and $c(2)$ compose a code vector, defined as $\underline{c} = [c(1)\ c(2)]^T$. During the first time slot, the two symbols $[c(1)\ c(2)]$ in the first row of $\mathbf{G}$ are transmitted simultaneously over the two transmit antennas. During the second time slot, the two symbols

$[-c^*(2)\ c^*(1)]$ of the second row of $\mathbf{G}$ are then transmitted. The transmission process of this 2×2 Alamouti scheme is illustrated in Fig. 2.3.

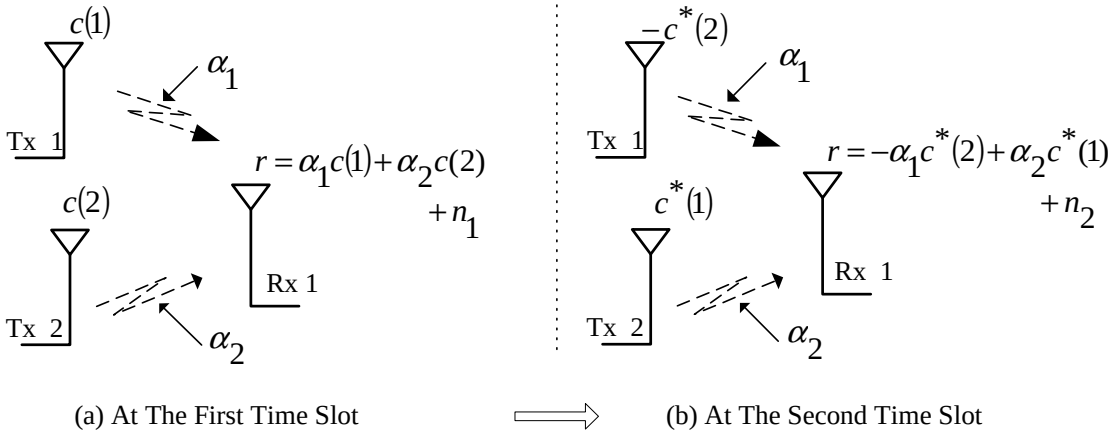


Figure 2.3 Illustration of 2×2 Alamouti STBC scheme.

The received signal at the mth receive antenna over two time slots can be written as

$$\begin{bmatrix} r_m(1) \\ r_m(2) \end{bmatrix} = \begin{bmatrix} c(1) & c(2) \\ -c^*(2) & c^*(1) \end{bmatrix} \begin{bmatrix} \alpha_{m,1} \\ \alpha_{m,2} \end{bmatrix} + \begin{bmatrix} n_m(1) \\ n_m(2) \end{bmatrix} \tag{2.5}$$

where $n_m(i), i = 1, 2$, is circularly symmetric complex Gaussian noise in the ith time slot and at the mth receive antenna, $\alpha_{m,i}$ is the fading gain from the ith transmit antenna to the mth receive antenna. With the quasi-static fading assumption of the wireless channels, $\alpha_{m,1}$ and $\alpha_{m,2}$ are constant over two time slots.

For notational convenience, (2.5) can be written in an alternative form as follows:

$$\underbrace{\begin{bmatrix} r_m(1) \\ r_m^*(2) \end{bmatrix}}_{\underline{r}_m} = \underbrace{\begin{bmatrix} \alpha_{m,1} & \alpha_{m,2} \\ \alpha_{m,2}^* & -\alpha_{m,1}^* \end{bmatrix}}_{\mathbf{H}_m} \underbrace{\begin{bmatrix} c(1) \\ c(2) \end{bmatrix}}_{\underline{c}} + \underbrace{\begin{bmatrix} n_m(1) \\ n_m^*(2) \end{bmatrix}}_{\underline{n}_m} \tag{2.6}$$

In the above expression, $\mathbf{H}_m$ is the equivalent channel response matrix corresponding to the transmission of $\underline{c}$ from $N = 2$ transmit antennas to the mth receive antenna over one block duration (i.e., P time slots). Observe that the elements of $\mathbf{H}_m$ depend not only on the channel fading coefficients of the transmit-receive antenna pairs, but also on the code constraint described by matrix $\mathbf{G}$.

Furthermore, to illustrate conveniently the advantage of the Alamouti scheme over the conventional system with only one transmit-receive antenna pair, assume that only one receive antenna, (i.e., $M_R = 1$) is employed for the simplest system with only one user. Compared to the conventional system, two symbols are now transmitted over two symbol times instead of one symbol over one symbol time. Although the Alamouti scheme offers the same data rate as that of the conventional scheme, it performs better in terms of BER.

Multiplying both sides of (2.6) by $\mathbf{H}_m^H$, where $(\cdot)^H$ denotes the Hermitian transpose operation, an equivalent form of the received signals corresponding to the transmitted symbols $c(1)$ and $c(2)$ is obtained as:

$$\begin{bmatrix} y_m(1) \\ y_m(2) \end{bmatrix} = \begin{bmatrix} \|\alpha_{m,1}\|^2 + \|\alpha_{m,2}\|^2 & 0 \\ 0 & \|\alpha_{m,1}\|^2 + \|\alpha_{m,2}\|^2 \end{bmatrix} \begin{bmatrix} c(1) \\ c(2) \end{bmatrix} + \begin{bmatrix} w_m(1) \\ w_m(2) \end{bmatrix} \tag{2.7}$$

where

$$\begin{bmatrix} y_m(1) \\ y_m(2) \end{bmatrix} = \begin{bmatrix} \alpha_{m,1}^* & \alpha_{m,2} \\ \alpha_{m,2}^* & -\alpha_{m,1} \end{bmatrix} \begin{bmatrix} r_m(1) \\ r_m^*(2) \end{bmatrix} \tag{2.8}$$

and

$$\begin{bmatrix} w_m(1) \\ w_m(2) \end{bmatrix} = \begin{bmatrix} \alpha_{m,1}^* & \alpha_{m,2} \\ \alpha_{m,2}^* & -\alpha_{m,1} \end{bmatrix} \begin{bmatrix} n_m(1) \\ n_m^*(2) \end{bmatrix} \tag{2.9}$$

It is important to point out here that the noise $w_m(1)$ and $w_m(2)$ are still independent circularly symmetric Gaussian random variables because the matrix $\mathbf{H}_m$ is unitary. This fact implies that the detection problem for $c(1)$ and $c(2)$ in (2.7) can be decomposed into two separate, scalar problems. Then, for one symbol, for example, $c(1)$, the received symbol can be written as:

$$y_m(1) = (\|\alpha_{m,1}\|^2 + \|\alpha_{m,2}\|^2)c(1) + w_m(1) \tag{2.10}$$

It follows from (2.9) and the fact that $n_m(i) \sim \mathcal{CN}(0, N_0)$, the distribution of $w_m(1)$ is

$$w_m(1) \sim \mathcal{CN}(0, (\|\alpha_{m,1}\|^2 + \|\alpha_{m,2}\|^2)N_0) \tag{2.11}$$

For a given signal-to-noise ratio (SNR), how well the symbols $c(1)$ and $c(2)$ can be detected correctly depends on the size of the coefficient $\|\alpha_{m,1}\|^2 + \|\alpha_{m,2}\|^2$, which is a summation of two independent channels' fading gain. For a fair comparison, first, SNR is defined as the value of the signal-to-noise ratio when the signal is transmitted over an AWGN channel. Second, assume that the total transmit power is the same for the conventional system and the system with Alamouti scheme.

Given the vector of channel fading gains, for the conventional system over fading channel, the actual signal-to-noise ratio at the receiver would be $\|\alpha_{m,1}\|^2$SNR since there is only one channel. For the systems with Alamouti scheme, the actual signal to noise ratio at the receiver would be $\frac{(\|\alpha_{m,1}\|^2+\|\alpha_{m,2}\|^2)}{2}$SNR. The factor $\frac{1}{2}$ for the Alamouti scheme comes from the fact that for the Alamouti scheme two symbols are transmitted simultaneously over each time period, so each symbol should be transmitted with half the power in the conventional system.

For a Rayleigh fading channel with the channel gain $\alpha_{m,i} \sim \mathcal{CN}(0,1)$, $\|\alpha_{m,i}\|^2$ is Chi-square distributed with 2 degrees of freedom. Therefore $\|\alpha_{m,1}\|^2 + \|\alpha_{m,2}\|^2$ is Chi-square distributed with 4 degrees of freedom. The higher order of the Chi-square distribution means that the tail of the distribution near zero becomes smaller. This translates to a lower probability of the events where the average fading gain becomes very small, thus leading to a higher chance to correctly detect the transmitted symbol.

To understand this better, one can examine the probability of the deep fade event in which the overall channel gain is small. For Alamouti scheme, this typical error event at high SNR happens with the following probability [5]:

$$P_1 = P\left\{\frac{(\|\alpha_{m,1}\|^2 + \|\alpha_{m,2}\|^2)}{2} < 1/\text{SNR}\right\} \approx \frac{2}{\text{SNR}^2} \tag{2.12}$$

On the other hand, for a conventional system, this probability is

$$P_2 = P\left\{\|\alpha_{m,1}\|^2 < 1/\text{SNR}\right\} \approx \frac{1}{\text{SNR}} \tag{2.13}$$

At high SNR, P_1 is much less than P_2. This clearly shows that with Alamouti scheme, the system can achieve a better detection performance. The order of the SNR at the

denominator in (2.12) and (2.13) is called the diversity gain of the system and it is denoted by L. As seen from above analysis, for Alamouti scheme $L = 2$, and for the conventional case $L = 1$. The higher order of the diversity gain L means that the probability of overall gain being small is lower, promising a better BER performance.

Now, return to the multiuser STBC system described in Fig. 2.1. Since all users in the system transmit over the wireless channel at the same time, their transmitted signals are superimposed at each receive antenna. However, with the well-designed structure of the orthogonal space-time block code, symbols belong to the same user do not interfere. Only the symbols belonging to different users interfere, creating the so-called inter-user interference. In particular, the combined received signal at the mth antenna is given as:

$$\begin{aligned} \underline{r}^m &= \sum_{k=1}^{K} \underline{r}_{mk} + \underline{n}^m \\ &= \left[\begin{array}{ccccccc} \mathbf{H}_{m1} & \mathbf{H}_{m2} & \cdots & \mathbf{H}_{mk} & \cdots & \mathbf{H}_{mK} \end{array} \right] \\ &\quad \cdot \left[\begin{array}{ccccccc} \underline{c}_1^T & \underline{c}_2^T & \cdots & \underline{c}_k^T & \cdots & \underline{c}_K^T \end{array} \right]^T + \underline{n}^m \end{aligned} \tag{2.14}$$

where $\underline{c}_k \triangleq [c_k(1), c_k(2), \cdots, c_k(P)]^T$ consists of the the kth user's transmitted symbols from time slots 1 to P. The vector $\underline{n}^m \triangleq [n^m(1), n^m(2), \ldots, n^m(P)]^T$ contains the additive Gaussian noise samples from time slots 1 to P at the mth receive antenna. $\mathbf{H}_{mk}$ is the kth user's equivalent channel response matrix corresponding to the transmission of $\underline{c}_k$ from its N transmit antennas to the mth receive antenna over one block duration (i.e., P time slots). Each element of $\mathbf{H}_{mk}$ is determined by the channel gain (modeled as a circularly symmetric complex Gaussian random variable of unit variance) and the space-time code used by the kth user.

Furthermore, from (2.14), it is not difficult to write out the mathematical model for the system equipped with M_R receive antennas. Specifically, the discrete-time

baseband received signal can be expressed as:

$$\underbrace{\begin{bmatrix} \underline{r}^1 \\ \vdots \\ \underline{r}^m \\ \vdots \\ \underline{r}^{M_R} \end{bmatrix}}_{\mathbf{r}_{M_R P\times 1}} = \underbrace{\begin{bmatrix} \mathbf{H}_{11}\cdots\mathbf{H}_{1k}\cdots\mathbf{H}_{1K} \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ \mathbf{H}_{m1}\cdots\mathbf{H}_{mk}\cdots\mathbf{H}_{mK} \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ \mathbf{H}_{M_R 1}\cdots\mathbf{H}_{M_R k}\cdots\mathbf{H}_{M_R K} \end{bmatrix}}_{\mathbf{H}_{M_R P\times NK}} \underbrace{\begin{bmatrix} \underline{c}_1 \\ \vdots \\ \underline{c}_k \\ \vdots \\ \underline{c}_K \end{bmatrix}}_{\mathbf{c}_{NK\times 1}} + \underbrace{\begin{bmatrix} \underline{n}^1 \\ \vdots \\ \underline{n}^m \\ \vdots \\ \underline{n}^{M_R} \end{bmatrix}}_{\mathbf{n}_{M_R P\times 1}} \tag{2.15}$$

where $\underline{r}^m \triangleq [r^m(1), r^m(2), \cdots, r^m(P)]^T$, consists of the received signals from time slots 1 to P at the mth receive antenna.

Based on the sufficient statistics in (2.15), the optimum receiver could be implemented. However, with the use of the maximal likelihood (ML) scheme in the front-end for STBC decoding and MUD, coupled with the decoding of the convolutional code for each user, the overall complexity of this method is prohibitive in a multiuser STBC system. In [9], a practical suboptimal scheme is presented by employing the iterative processing technique for joint detection and decoding. This implementation is referred to as the conventional iterative receiver in this thesis. The detailed implementation of this receiver is discussed in the following section.

2.2 Conventional Iterative Receiver

Fig. 2.4 illustrates the conventional iterative receiver for a multiuser STBC system. It consists of a soft-output multiuser detector, followed by K parallel demodulators and channel decoders. The demodulator and the decoder in one branch are separated by the interleaver and deinterleaver. The MUD takes as its input the received signals from the M_R receive antennas and the interleaved extrinsic log-likelihood ratios (LLR's) of the coded bits of all users $\{\lambda_2^{\pi}[d_k(l,j)]\}$, which are fed back from K users' MAP channel decoders. The definition of LLR was made in (2.42). The MUD computes the soft estimate $\hat{c}_k(l)$ of the kth user's lth channel symbol $c_k(l)$, and then feeds them into K single-user MAP demodulators. The demodulator takes as its input both the MMSE estimate $\hat{c}_k(l)$ of symbol $c_k(l)$ from the MUD, and again the interleaved

extrinsic LLR's $\{\lambda_2^{\pi}[d_k(l,j)]\}$ of the corresponding coded bits from the MAP channel decoder. Then the demodulator computes the *a posteriori* LLR's $\{\Lambda_1[d_k(l,j)]\}$ of the coded bits with the MAP algorithm [11]. The MAP channel decoder of each user takes the interleaved extrinsic LLR's of the coded bits $\{\lambda_1^{\pi}[d_k(l,j)]\}$ from the corresponding demodulator and computes the *a posterior* LLR's $\{\lambda_2[d_k(l,j)]\}$ of the coded bits and the LLR's $\{\Lambda_2[b_k(i)]\}$ of the information bits based on the structure of the convolutional code.

In [9], an iterative receiver algorithm is given for the case when M-PSK constellation is adopted in the system, where each channel symbol is transmitted with the same energy. Here the algorithm is extended to a more general case with M-QAM constellation in which the symbols of the constellation have different energies. In the following subsections, emphasis is placed on the soft-output MUD and the MAP demodulator. For the MAP channel decoder, only the most significant features are highlighted and discussed to help with the complexity analysis. More details of the MAP channel decoder can be found in [11].

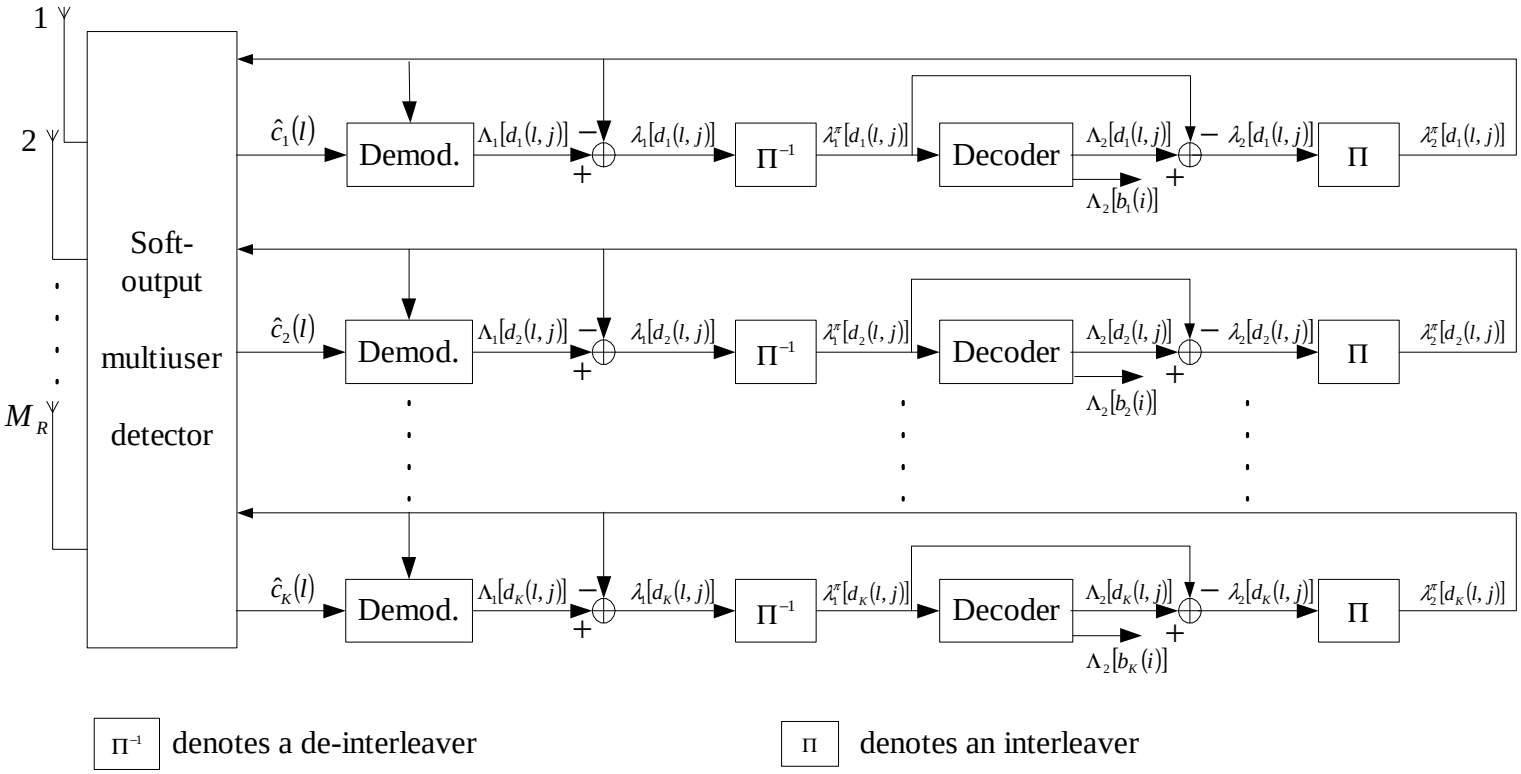


Figure 2.4 Conventional iterative receiver for a multiuser STBC system.

2.2.1 Soft-Output MUD with Interference Cancellation

The basic idea of the soft-output MUD with interference cancellation is to subtract the soft estimates of the interference symbols from the received signal and then apply the instantaneous linear MMSE filter to the residue signal to obtain a better estimate of the transmitted symbol.

First the *soft* estimate $\tilde{c}_k(l)$ of the kth user's lth code symbol $c_k(l)$ and the estimated $E\{\|c_k(l)\|^2\}$ of its energy are defined as follows:

$$\tilde{c}_k(l) \triangleq E[c_k(l)] = \sum_{C_i \in \Omega_C} C_i \, P[c_k(l) = C_i] \tag{2.16}$$

$$E\{\|c_k(l)\|^2\} \triangleq \sum_{C_i \in \Omega_C} \|C_i\|^2 \, P[c_k(l) = C_i] \tag{2.17}$$

where Ω_C denotes the symbol constellation. The adjective *'soft'* comes from the fact that the probability $P[c_k(l) = C_i]$ is calculated based on the extrinsic LLR's of the coded bits carried by symbol $c_k(l)$.

At the first iteration, no *a priori* information about the code symbols is avaliable, thus the code symbols are assumed to be equiprobable, i.e., $P[c_k(l) = C_i] = 1/|\Omega_C|$. In the subsequent iterations, the probability $P[c_k(l) = C_i]$ can be computed from the feedback extrinsic information delivered by the channel decoder, as will be explained in the next section.

For the K-user STBC system described in (2.15), define an (NK)-dimensional *soft* vector corresponding to the true transmitted symbol vector $\mathbf{c} = [\underline{c}_1^T, \underline{c}_2^T, \cdots, \underline{c}_K^T]^T$ as:

$$\begin{aligned} \tilde{\mathbf{c}} &\triangleq [\underline{\tilde{c}}_1^T, \underline{\tilde{c}}_2^T, \cdots, \underline{\tilde{c}}_K^T]^T \\ &= [\tilde{c}_1(1), \cdots, \tilde{c}_1(N), \cdots, \tilde{c}_K(1), \cdots, \tilde{c}_K(N)]^T \end{aligned} \tag{2.18}$$

In order to apply the techniques proposed in [11], one can treat every element in $\tilde{\mathbf{c}}$ as belonging to a virtual user. Therefore the system in (2.15) with K actual users can be treated as a system with NK virtual users. Define

$$\tilde{\mathbf{c}}_k(l) \triangleq \tilde{\mathbf{c}} - \tilde{c}_k(l)\tilde{\mathbf{e}}_k(l) \tag{2.19}$$

In (2.19) and the following, the combination of (k, l) is used to index a virtual user, and $\tilde{\mathbf{e}}_k(l)$ is an (NK)-vector of all zeros, except for the "1" element in the corresponding entry of the (k, l)th virtual user. That is, $\tilde{\mathbf{c}}_k(l)$ is obtained from $\tilde{\mathbf{c}}$ by setting the (k, l)th element to zero.

Subtracting the soft estimates of the interfering signals of other virtual users from the received signal $\mathbf{r}$ in (2.15), gives

$$\tilde{\mathbf{r}}_k(l) \triangleq \mathbf{r} - \mathbf{H}\tilde{\mathbf{c}}_k(l) = \mathbf{H}[\mathbf{c} - \tilde{\mathbf{c}}_k(l)] + \mathbf{n} \tag{2.20}$$

In order to further suppress the multiuser interference in $\tilde{\mathbf{r}}_k(l)$, the instantaneous linear MMSE filter is applied to estimate the transmitted symbol. The linear MMSE weight vector $\mathbf{w}_k(l)$ is chosen to minimize the mean square error (MSE) between the transmitted symbol $c_k(l)$ and the filter output, which is expressed as

$$\hat{c}_k(l) \triangleq \mathbf{w}_k^H(l)\tilde{\mathbf{r}}_k(l) \tag{2.21}$$

where $(\cdot)^H$ denotes Hermitian transpose operation. Thus $\mathbf{w}_k(l)$ is found as

$$\begin{aligned} \mathbf{w}_k(l) &= \arg \underbrace{\min}_{\mathbf{w} \in \mathcal{C}^{MP}} E\left[||c_k(l) - \mathbf{w}^H\tilde{\mathbf{r}}_k(l)||^2\right] \\ &= E\{\tilde{\mathbf{r}}_k(l)\tilde{\mathbf{r}}_k^H(l)\}^{-1}E\{c_k^*(l)\tilde{\mathbf{r}}_k(l)\} \end{aligned} \tag{2.22}$$

Below shows how to compute $E\{c_k^*(l)\tilde{\mathbf{r}}_k(l)\}$ and $E\{\tilde{\mathbf{r}}_k(l)\tilde{\mathbf{r}}_k^H(l)\}$ in (2.22). First, the term $E\{c_k^*(l)\tilde{\mathbf{r}}_k(l)\}$ is computed as follows,

$$\begin{aligned} E\{c_k^*(l)\tilde{\mathbf{r}}_k(l)\} &= \mathbf{H}E\{c_k^*(l)\left[\mathbf{c} - \tilde{\mathbf{c}}_k(l)\right]\} \\ &= \mathbf{H}E\left\{c_k^*(l) \cdot \begin{bmatrix} c_1(1) - \tilde{c}_1(1) \\ \vdots \\ c_k(l-1) - \tilde{c}_k(l-1) \\ c_k(l) \\ c_k(l+1) - \tilde{c}_k(l+1) \\ \vdots \\ c_K(N) - \tilde{c}_K(N) \end{bmatrix}\right\} \end{aligned} \tag{2.23}$$

Using the fact that all the transmitted symbols are independent, and the definition in (2.16), one obtains:

$$E\{[c_k(l) - \tilde{c}_k(l)]\} = E\{c_k(l)\} - \tilde{c}_k(l) = \tilde{c}_k(l) - \tilde{c}_k(l) = 0 \tag{2.24}$$

Now substituting (2.24) into (2.23) yields:

$$\begin{aligned} E\{c_k^*(l)\tilde{\mathbf{r}}_k(l)\} &= \mathbf{H}\begin{bmatrix} E\{c_k^*(l)\cdot[c_1(1)-\tilde{c}_1(1)]\} \\ \vdots \\ E\{c_k^*(l)\cdot[c_k(l-1)-\tilde{c}_k(l-1)]\} \\ E\{c_k^*(l)\cdot c_k(l)\} \\ E\{c_k^*(l)\cdot[c_k(l+1)-\tilde{c}_k(l+1)]\} \\ \vdots \\ E\{c_k^*(l)\cdot[c_K(N)-\tilde{c}_K(N)]\} \end{bmatrix} = \mathbf{H}\begin{bmatrix} E\{c_k^*(l)\}\cdot 0 \\ \vdots \\ E\{c_k^*(l)\}\cdot 0 \\ E\{c_k^*(l)\cdot c_k(l)\} \\ E\{c_k^*(l)\}\cdot 0 \\ \vdots \\ E\{c_k^*(l)\}\cdot 0\} \end{bmatrix} \\ &= \mathbf{H}\cdot\tilde{\mathbf{e}}_k(l)\cdot E\left\{||c_k(l)||^2\right\} \end{aligned} \tag{2.25}$$

Next, the term $E\{\tilde{\mathbf{r}}_k(l)\tilde{\mathbf{r}}_k^H(l)\}$ in (2.22) is computed as follows,

$$\begin{aligned} E\{\tilde{\mathbf{r}}_k(l)\tilde{\mathbf{r}}_k^H(l)\} &= E\left\{[\mathbf{H}(\mathbf{c}-\tilde{\mathbf{c}}_k(l))+\mathbf{n}][\mathbf{H}(\mathbf{c}-\tilde{\mathbf{c}}_k(l))+\mathbf{n}]^H\right\} \\ &= E\left\{[\mathbf{H}(\mathbf{c}-\tilde{\mathbf{c}}_k(l))][\mathbf{H}(\mathbf{c}-\tilde{\mathbf{c}}_k(l))]^H\right\} + E\left\{\mathbf{n}\mathbf{n}^H\right\} \\ &= \mathbf{H}E\left\{(\mathbf{c}-\tilde{\mathbf{c}}_k(l))(\mathbf{c}-\tilde{\mathbf{c}}_k(l))^H\right\}\mathbf{H}^H + E\left\{\mathbf{n}\mathbf{n}^H\right\} \\ &= \mathbf{H}\text{cov}[\mathbf{c}-\tilde{\mathbf{c}}_k(l)]\mathbf{H}^H + N_0\mathbf{I} \end{aligned} \tag{2.26}$$

where N_0 denotes the one-sided power spectrum density of the white Gaussian noise; $\mathbf{I}$ is the identity matrix; and $\text{cov}(\cdot)$ denotes the covariance operator. Define

$$\begin{aligned} \mathbf{V}_k(l) \triangleq \text{cov}[\mathbf{c}-\tilde{\mathbf{c}}_k(l)] = \text{diag}\{&E\{\|c_1(1)\|^2\} - \|\tilde{c}_1(1)\|^2, \cdots, \\ &E\{\|c_1(N)\|^2\} - \|\tilde{c}_1(N)\|^2, \cdots, E\{\|c_k(l-1)\|^2\} - \|\tilde{c}_k(l-1)\|^2, E\{\|c_k(l)\|^2\}, \\ &E\{\|c_k(l+1)\|^2\} - \|\tilde{c}_k(l+1)\|^2, \cdots, E\{\|c_K(N)\|^2\} - \|\tilde{c}_K(N)\|^2\} \end{aligned} \tag{2.27}$$

where $\text{diag}(\cdot)$ means a diagonal matrix whose diagonal elements are inside the parentheses and $E\{\|c_k(l)\|^2\}$ is calculated as in (2.17). Then (2.26) can be rewritten as

$$E\{\tilde{\mathbf{r}}_k(l)\tilde{\mathbf{r}}_k^H(l)\} = \mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H + N_0\mathbf{I} \tag{2.28}$$

In particular, for M-PSK, all the transmitted symbols have the same energy, i.e., $E\{\|c_k(l)\|^2\} = E_s$. Therefore, $\mathbf{V}_k(l)$ can be further simplified to

$$\begin{aligned}\mathbf{V}_k(l) = \text{diag}\{&E_s - \|\tilde{c}_1(1)\|^2, \cdots, E_s - \|\tilde{c}_1(N)\|^2, \cdots, \\ &E_s - \|\tilde{c}_k(l-1)\|^2, E_s, E_s - \|\tilde{c}_k(l+1)\|^2, \cdots, E_s - \|\tilde{c}_K(N)\|^2\}\end{aligned} \tag{2.29}$$

Now, substituting (2.25), (2.28) and (2.22) into (2.21), the final expressions of $\mathbf{w}_k(l)$ and $\hat{c}_k(l)$ for a general M-QAM constellation are obtained as:

$$\mathbf{w}_k(l) \;=\; [\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H + N_0\mathbf{I}]^{-1}[\mathbf{H}\cdot\tilde{\mathbf{e}}_k(l)\cdot E\left\{||c_k(l)||^2\right\}] \tag{2.30}$$

and

$$\begin{aligned}\hat{c}_k(l) &\triangleq \mathbf{w}_k^H(l)\tilde{\mathbf{r}}_k(l) \\ &= \{[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H + N_0\mathbf{I}]^{-1}[\mathbf{H}\cdot\tilde{\mathbf{e}}_k(l)\cdot E\left\{||c_k(l)||^2\right\}]\}^H\cdot\tilde{\mathbf{r}}_k(l) \\ &= \tilde{\mathbf{e}}_k^H(l)\cdot\mathbf{H}^H\cdot[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H + N_0\mathbf{I}]^{-1}\cdot\tilde{\mathbf{r}}_k(l)\cdot E\left\{||c_k(l)||^2\right\}\end{aligned} \tag{2.31}$$

2.2.2 Gaussian Approximation for the Output of the MMSE Filter

The instantaneous output of the MMSE filter can be modeled as the output of an equivalent AWGN channel having $c_k(l)$ as its input symbol [9]. The simulation results in the following chapters will also show that such an assumption is reasonable and quite accurate. This assumption greatly simplifies the computation of the *a posterior* probability of the estimated symbol. With this assumption, the output of the MMSE filter can be represented as

$$\hat{c}_k(l) = \mu_k(l)c_k(l) + \nu_k(l) \tag{2.32}$$

where the parameters $\mu_k(l)$ and $E\{\nu_k^2(l)\}$ can be determined as follows.

First, multiply both sides of (2.32) with $c_k^*(l)$ and find the expectation of $E\{\hat{c}_k(l)c_k^*(l)\}$

as

$$\begin{aligned} E\{\hat{c}_k(l)c_k^*(l)\} &= E\{[\mu_k(l)c_k(l)+\nu_k(l)]\cdot c_k^*(l)\} \\ &= E\{\mu_k(l)c_k(l)c_k^*(l)+\nu_k(l)c_k^*(l)\} \\ &= \mu_k(l)\cdot E\{c_k(l)c_k^*(l\}+E\{\nu_k(l)c_k^*(l)\} \\ &= \mu_k(l)E\{\|c_k(l)\|^2\} \end{aligned} \tag{2.33}$$

Second, multiply both sides of (2.31) and compute $E\{\hat{c}_k(l)c_k^*(l)\}$ as

$$\begin{aligned} E\{\hat{c}_k(l)c_k^*(l)\} &= E\left\{\tilde{\mathbf{e}}_k^H(l)\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+N_0\mathbf{I}]^{-1}\tilde{\mathbf{r}}_k(l)E\left\{\|c_k(l)\|^2\right\}\right\}c_k^*(l) \\ &= E\{\|c_k(l)\|^2\}\tilde{\mathbf{e}}_k^H(l)\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+N_0\mathbf{I}]^{-1}E\{\tilde{\mathbf{r}}_k(l)c_k^*(l)\} \end{aligned} \tag{2.34}$$

Substituting (2.25) into (2.34), $E\{\hat{c}_k(l)c_k^*(l)\}$ can be further transformed to

$$\begin{aligned} E\{\hat{c}_k(l)c_k^*(l)\} &= E\{\|c_k(l)\|^2\}\tilde{\mathbf{e}}_k^H(l)\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+N_0\mathbf{I}]^{-1}E\{\tilde{\mathbf{r}}_k(l)c_k^*(l)\} \\ &= E\{\|c_k(l)\|^2\}\tilde{\mathbf{e}}_k^H(l)\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+N_0\mathbf{I}]^{-1}\mathbf{H}\tilde{\mathbf{e}}_k(l)E\{\|c_k(l)\|^2\} \end{aligned} \tag{2.35}$$

By comparison of (2.33) and (2.35), it is deduced that

$$\mu_k(l) \triangleq \{\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+\sigma^2\mathbf{I}]^{-1}\mathbf{H}\}_{kk}\cdot E\{\|c_k(l)\|^2\} \tag{2.36}$$

Now the expression of $E\{\nu_k^2(l)\}$ is derived as:

$$E\{\nu_k^2(l)\} \triangleq \mathrm{var}\{\hat{c}_k(l)\} = E\{\|\hat{c}_k(l)\|^2\}-\mu_k^2(l)E\{\|c_k(l)\|^2\} \tag{2.37}$$

From $\hat{c}_k(l) \triangleq \mathbf{w}_k^H(l)\tilde{\mathbf{r}}_k(l)$, (2.28) and (2.30), $E\{\|\hat{c}_k(l)\|^2\}$ can be expressed as

$$\begin{aligned} E\{\|\hat{c}_k(l)\|^2\} &= E\{\mathbf{w}_k^H(l)\tilde{\mathbf{r}}_k(l)[\mathbf{w}_k^H(l)\tilde{\mathbf{r}}_k(l)]^H\} = \mathbf{w}_k^H(l)E\{\tilde{\mathbf{r}}_k(l)\tilde{\mathbf{r}}_k(l)^H\}\mathbf{w}_k(l) \\ &= \mathbf{w}_k^H(l)\{\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+N_0\mathbf{I}\}\mathbf{w}_k(l) \\ &= E\{\|c_k(l)\|^2\}\tilde{\mathbf{e}}_k^H(l)\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+N_0\mathbf{I}]^{-1}\mathbf{H}\tilde{\mathbf{e}}_k(l)E\{\|c_k(l)\|^2\} \\ &= E\{\|c_k(l)\|^2\}\{\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H+\sigma^2\mathbf{I}]^{-1}\mathbf{H}\}_{kk}\cdot E\{\|c_k(l)\|^2\} \\ &= E\{\|c_k(l)\|^2\}\cdot\mu_k(l) \end{aligned} \tag{2.38}$$

It then follows that

$$E\{\nu_k^2(l)\} = [\mu_k(l)-\mu_k^2(l)]E\{\|c_k(l)\|^2\} \tag{2.39}$$

Based on the Gaussian model of (2.32), (2.36), and (2.39), the *a posterior* probability of the estimated symbol can be computed as

$$P[\hat{c}_k(l)|c_k(l) = C_i] = \frac{1}{\pi E\{\nu_k^2(l)\}} \cdot \exp\left(-\frac{\|\hat{c}_k(l) - \mu_k(l)C_i\|^2}{E\{\nu_k^2(l)\}}\right) \tag{2.40}$$

where $i = 1, 2, \cdots, M$ and $C_i \in \Omega_C$.

2.2.3 MAP Demodulator

Suppose that every complex symbol $c_k(l)$ carries a J-dimensional vector as

$$c_k(l) \triangleq [d_k(l,1), \cdots, d_k(l,j), \cdots, d_k(l,J)]^T \tag{2.41}$$

where $J = \log_2 M$ and $d_k(l,j) \in \{+1, -1\}$ denotes the jth labelling bit of the kth user's lth complex symbol. The LLR value of one coded bit $\lambda[d_k(l,j)]$ is defined as

$$\lambda[d_k(l,j)] \triangleq \log \frac{P[d_k(l,j) = +1]}{P[d_k(l,j) = -1]} \tag{2.42}$$

The demodulator takes as its input both the MMSE estimate $\hat{c}_k(l)$ of symbol $c_k(l)$ and the interleaved extrinsic LLR's $\{\lambda_2^{\pi}[d_k(l,j)]\}$ where the superscript π means that it is the interleaved extrinsic LLR. It then computes the *a posteriori* LLR's $\{\Lambda_1[d_k(l,j)]\}$ of the coded bits with the MAP algorithm as follows:

$$\begin{aligned}
\Lambda_1[d_k(l,j)] &\triangleq \log \frac{P[d_k(l,j) = 1|\hat{c}_k(l)]}{P[d_k(l,j) = 0|\hat{c}_k(l)]} \\
&= \log \frac{P[\hat{c}_k(l)|d_k(l,j) = 1]P[d_k(l,j) = 1]}{P[\hat{c}_k(l)|d_k(l,j) = -1]P[d_k(l,j) = -1]} \\
&= \log \frac{\sum_{C_i \in C_j^+} P[\hat{c}_k(l)|c_k(l) = C_i]P[C_i]}{\sum_{C_i \in C_j^-} P[\hat{c}_k(l)|c_k(l) = C_i]P[C_i]}
\end{aligned} \tag{2.43}$$

$$P[C_i] = \prod_{j=1}^{J} P[d_k(l,j) = D(i,j)] \tag{2.44}$$

where $D(i,j) \in \{+1, -1\}$ is the jth labelling bit of C_i, C_j^+ and C_j^- are the sets of the complex symbols whose jth labelling bits are "1" and "-1", respectively.

Subtracting the *a priori* information $\lambda_2^{\pi p}[d_k(l,j)]$ of the coded bits from the MAP-decoder, the *extrinsic* LLR's of $d_k(l,j)$ at the output of the MAP demodulator can

be obtained as

$$\lambda_1[d_k(l,j)] = \Lambda_1[d_k(l,j)] - \lambda_2^{\pi p}[d_k(l,j)] \tag{2.45}$$

where the letter p in the superscript of $\lambda_2^{\pi p}[d_k(l,j)]$ indicates that the information is from the previous iteration. The quantity $\lambda_2^{\pi p}[d_k(l,j)]$ is computed by subtracting the LLR of the coded bit at the input of the MAP-decoder from the corresponding LLR at the output. At the first iteration, no *a priori* information about the coded bits is available, thus $\lambda_2^{\pi p}[d_k(l,j)] = 0$. Finally the extrinsic LLR's calculated in (2.45) are deinterleaved, and then fed to the MAP decoder.

According to the definition of the LLR in (2.42), it is not difficult to convert the LLR value into the corresponding probabilities as follows:

$$P[d_k(l,j) = +1] = \frac{\exp\left[\lambda[d_k(l,j)]\right]}{1 + \exp\left[\lambda[d_k(l,j)]\right]} \tag{2.46}$$

$$P[d_k(l,j) = -1] = \frac{1}{1 + \exp\left[\lambda[d_k(l,j)]\right]} \tag{2.47}$$

2.2.4 Channel Decoder

The channel decoder for the convolutional code is implemented separately for each user with the MAP decoding algorithm. The MAP channel decoder of the kth user takes the interleaved extrinsic LLR's of the coded bits $\{\lambda_1^{\pi}[d_k(l,j)]\}$ from the corresponding demodulator and computes the *a posterior* LLR's $\{\lambda_2[d_k(l,j)]\}$ of the coded bits and the LLR's $\{\Lambda_2[b_k(i)]\}$ of the information bits based on the structure of the convolutional code [13].

The basic concepts of encoding and MAP decoding for convolutional codes are presented in Appendix A. This section outlines a procedure for computing the LLR's of the information and coded bits for a rate-$\frac{k_0}{n_0}$ convolutional code of overall constraint length $k_0\nu$. It will also help to analyze and demonstrate the algorithm complexity in Chapter 3. More details of the algorithm can also be found in [11].

The input to the encoder at time t is the block $\underline{d}_t = (d_t^1, \cdots, d_t^{k_0})$ and the corresponding output of the encoder is $\underline{b}_t = (b_t^1, \cdots, b_t^{n_0})$. The state of the trellis at

time t can be represented by a $k_0(\nu-1)$-tuple, as $S_t = (s_t^1, \cdots, s_t^{k_0(\nu-1)})$. Denote the input information bits that cause the state transition from $S_{t-1} = s'$ to $S_t = s$ by $\underline{d}(s', s)$ and the corresponding output coded bits by $\underline{b}(s', s)$. The encoder starts in state S_0 (S_0 is always a all-zero state). An information bit stream $\{\underline{d}_t\}_{t=1}^T$ is fed into the encoder, followed by ν blocks of all zero bits, forcing the encoder to end in all-zero state again at time $\tau = T + \nu$, i.e., $S_\tau = S_0$.

As in Appendix A, define

$$P[b_t(s', s)] \triangleq P[\underline{b}_t = b_t(s', s)] \tag{2.48}$$

and the following forward and backward recursions as:

$$\alpha_t(s) = \sum_{s'} \alpha_{t-1}(s') P[\underline{b}_t(s', s)],\ t = 1, 2, \cdots, \tau \tag{2.49}$$

$$\beta_t(s) = \sum_{s'} \beta_{t+1}(s') P[\underline{b}_{t+1}(s', s)],\ t = \tau-1, \tau-2, \cdots, 0 \tag{2.50}$$

with boundary conditions $\alpha(0) = 1$, $\alpha(s \neq 0) = 0$; and $\beta_\tau(0) = 1$, $\beta_\tau(s \neq 0) = 0$. In (2.49) and (2.50) the summations are over all states s' where the transition (s', s) is possible.

Let S_j^+ be the set of state pairs (s', s) such that the jth bit of $\underline{b}(s', s)$ is $+1$. Similarly, define S_j^{-1} as the set of state pairs (s', s) such that the jth bit of $\underline{b}(s', s)$ is -1. The *a posteriori* LLR of the coded bit b_t^j at the output of the MAP channel decoder can be computed as:

$$\begin{aligned} \Lambda_2[b_t^j] &\triangleq \log \frac{P[b_t^j = +1|\text{decoding}]}{P[b_t^j = -1|\text{decoding}]} \\ &= \underbrace{\log \frac{\sum_{S_j^+} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i \neq j} P[b_t^i(s', s)]}{\sum_{S_j^-} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i \neq j} P[b_t^i(s', s)]}}_{\lambda_2[b_t^j]} + \underbrace{\log \frac{P[b_t^j = +1]}{P[b_t^j = -1]}}_{\lambda_1^p[b_t^j]} \end{aligned} \tag{2.51}$$

It is seen from (2.51) that the output of MAP channel decoder is the sum of the *a priori* information $\lambda_1^p[b_t^j]$ provided by the multiuser detector and the extrinsic information $\lambda_2[b_t^j]$ based on the trellis structure of the code.

However, a direct implementation of the recursions (2.49) and (2.50) is numerically unstable, since $\alpha(s)$ and $\beta(s)$ drop toward zero exponentially. In order to obtain a numerically stable algorithm, these quantities must be scaled as the computation proceeds [11]. Let $\tilde{\alpha}_t(s)$ denote the scaled version of $\alpha_t(s)$. Initially, $\alpha_1(s)$ is computed as (2.49). By setting $\hat{\alpha}_1(s) = \alpha_1(s)$ and $\tilde{\alpha}_1 = c_1\hat{\alpha}_1(s)$ with $c_1 \triangleq 1/\sum_s \hat{\alpha}_1(s)$, for each $t \geq 2$, $\tilde{\alpha}_t(s)$ can be computed as:

$$\hat{\alpha}_t(s) = \sum_{s'} \tilde{\alpha}_{t-1}(s')P[\underline{b}_t(s', s)] \tag{2.52}$$

$$\tilde{\alpha}_t(s) = c_t\hat{\alpha}_t(s), \text{ where } c_t = 1/\sum_s \hat{\alpha}_t(s) \tag{2.53}$$

By a simple induction, one obtains $\tilde{\alpha}_{t-1}(s) = (\prod_{i=1}^{t-1} c_i)\alpha_{t-1}(s) \triangleq C_{t-1}\alpha_{t-1}(s)$. Thus $\tilde{\alpha}_t(s)$ can be rewritten as

$$\tilde{\alpha}_t(s) = \frac{\sum_{s'} C_{t-1}\alpha_{t-1}(s')P[\underline{b}_t(s', s)]}{\sum_s \sum_{s'} C_{t-1}\alpha_{t-1}(s')P[\underline{b}_t(s', s)]} = \frac{\alpha_t(s)}{\sum_s \alpha_t(s)} \tag{2.54}$$

That is, each $\alpha_t(s)$ is effectively scaled by the sum over all states of $\alpha_t(s)$.

Similarly, let $\tilde{\beta}_t(s)$ denote the scaled version of $\beta_t(s)$. By setting $\hat{\beta}_{\tau-1}(s) = \beta_{\tau-1}(s)$, for each $t < \tau - 1$, $\tilde{\beta}_t(s)$ can be computed as:

$$\hat{\beta}_t(s) = \sum_{s'} \tilde{\beta}_{t+1}(s')P[\underline{b}_{t+1}(s, s')] \tag{2.55}$$

$$\tilde{\beta}_t(s) = c_t\hat{\beta}_t(s) \tag{2.56}$$

where $\beta_{\tau-1}(s)$ is computed according to (2.50). With simple induction, one obtains that $\tilde{\beta}_t(s)$ is actually scaled by the product of $(\prod_{i=t}^{\tau} c_i)$ as

$$\tilde{\beta}_t(s) = \left(\prod_{i=t}^{\tau} c_i\right)\beta_t(s) \triangleq D_t\beta_t(s). \tag{2.57}$$

With (2.54) and (2.57), (2.51) can be written as

$$\Lambda_2[b_t^j] = \underbrace{\log \frac{\sum_{S_j^+} \tilde{\alpha}_{t-1}(s') \cdot \tilde{\beta}_t(s) \cdot \prod_{i \neq j} P[b_t^i(s', s)]}{\sum_{S_j^-} \tilde{\alpha}_{t-1}(s') \cdot \tilde{\beta}_t(s) \cdot \prod_{i \neq j} P[b_t^i(s', s)]}}_{\lambda_2[b_t^j]} + \underbrace{\log \frac{P[b_t^j = +1]}{P[b_t^j = -1]}}_{\lambda_1^p[b_t^j]} \tag{2.58}$$

which follows from the fact that $C_{t-1}D_t = \prod_{i=1}^{t-1} c_i \cdot \prod_{i=t}^{\tau} c_i = \prod_{i=1}^{\tau} c_i$ is a constant and it is independent of t.

The *a posteriori* LLR of the information bit can be computed in a similar way. Let $\mathcal{U}_j^+$ be the set of the state pairs (s', s) such that the jth bit of $\underline{d}(s', s)$ is $+1$. Similarly, $\mathcal{U}_j^{-1}$ is the set of the state pairs (s', s) such that the jth bit of $\underline{d}(s', s)$ is -1. Then

$$\Lambda_2[d_t^j] \;=\; \log \frac{\sum_{\mathcal{U}_j^+} \tilde{\alpha}_{t-1}(s') \cdot \tilde{\beta}_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s', s)]}{\sum_{\mathcal{U}_j^-} \tilde{\alpha}_{t-1}(s') \cdot \tilde{\beta}_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s', s)]} \tag{2.59}$$

Note that the computation of the LLR's of the information bits is only needed at the last iteration. The information bit d_t^j is then decoded as $\hat{d}_t^j = \mathrm{sgn}(\Lambda_2[d_t^j])$.

Finally, since the input to the MAP channel decoder is the LLRs of the coded bits, the probability $P[b_t^i(s', s)]$ can be expressed in terms of the LLR $\lambda_1^p[b_t^i]$ as [cf. (2.46) and (2.47)]:

$$P[b_t^i(s', s)] \;=\; \frac{\exp\left(b^i(s', s)\lambda_1^p[b^i]\right)}{1 + \exp\left(b^i(s', s)\lambda_1^p[b^i]\right)} \tag{2.60}$$

3. Complexity and Efficiency Analysis of the Iterative Receiver

3.1 System Complexity Analysis

In order to determine the system complexity and facilitate the comparison of different detection and demodulation schemes presented in this thesis, it is necessary to quantitatively measure the system complexity in terms of floating point operations (FLOP) [12]. Considering the big difference in computation between Multiplication/Division and Addition/Subtraction, we only take into account the Multiplication/Division operations and ignore the Addition/Subtraction operations. Furthermore, for complex numbers, multiplication and/or division operations are counted as four equivalent real-number Multiplication/Division operations. In addition, we count separately the exponential and logarithm operations because they consume much more resource than the simple multiplication or division operations in terms of memory or CPU time. In the following, MUL is used to denote one Multiplication/Division operation.

The next sections discuss the complexity of the channel decoder, the demodulator and the multiuser detector.

3.1.1 Complexity of the MAP Channel Decoder

First, the key parameters of the convolutional code are summarized as follows:

- The code rate is k_0/n_0
- The overall constraint length is $k_0\nu$

- The total possible transition states are $2^{k_0(\nu-1)}$

Next, let's determine how many operations that are needed to calculate the *a posteriori* probabilities of n_0 coded bits in one *typical* cycle. Here, a *typical* cycle means that it is not the few initial nor final cycles in one transmission block of the information bits. The following steps need to be carried out for the MAP channel decoder:

1. Equation (2.48):

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

   There are a total of 2^{n_0} possible vectors of output coded bits during the state transition. For each state transition (s', s), it takes $(n_0 - 1)$ MULs to calculate the probability $P[b_t(s', s)]$ based on the *soft* information of the n_0 coded bits. Therefore it takes $(n_0-1)\cdot 2^{n_0}$ MULs to calculate the corresponding probabilities of all the possible state transitions.

2. Equation (2.52):

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

   There are $2^{k_0(\nu-1)}$ possible states. For each state s, it takes 2^{k_0} MULs to calculate the corresponding value of $\hat{\alpha}_t(s)$ because there are only 2^{k_0} possible state transitions (s', s) for each specific state. Thus it takes $2^{k_0} \cdot 2^{k_0(\nu-1)}$ MULs to calculate all the values of $\hat{\alpha}_t(s)$ for the forward recursion.

3. Equation (2.53):

$$\tilde{\alpha}_t(s) \;=\; c_t \hat{\alpha}_t(s), \text{ with } c_t = 1/\sum_{s} \hat{\alpha}_t(s)$$

   The normalization of $\hat{\alpha}_t(s)$ to produce $\tilde{\alpha}_t(s)$ requires $(2^{k_0(\nu-1)} + 1)$ MULs for all the states.

4. Equation (2.55):

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

The calculation of $\hat{\beta}_t(s)$ is similar to the calculation of $\hat{\alpha}_t(s)$. So it takes $2^{k_0} \cdot 2^{k_0(\nu-1)}$ MULs for the backward recursion.

5. Equation (2.56)

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

The normalization of $\hat{\beta}_t(s)$ to obtain $\tilde{\beta}_t(s)$ also needs $2^{k_0(\nu-1)}$ MULs for all the states.

6. Equation (2.58):

$$\begin{aligned} \Lambda_2[b_t^j] &= \log \frac{\sum_{S_j^+} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s',s)]}{\sum_{S_j^-} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s',s)]} \\ &= \log \frac{\sum_{S_j^+} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot P[\underline{b}_t(s',s)]}{\sum_{S_j^-} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot P[\underline{b}_t(s',s)]}, \;\; j = 1, \cdots, n_0 \end{aligned} \tag{3.1}$$

Since $\alpha_{t-1}(s')$, $\beta_t(s)$ and $P[\underline{b}_t(s',s)]$ are already obtained from the previous steps. S_j^+ and S_j^- consist exactly of all $2^{k_0(\nu-1)} \cdot 2^{k_0}$ state transition possibilities. For each state transition (s',s), two MULs are needed to compute $\alpha_{t-1}(s')\beta_t(s)\prod_{i=1}^{n_0} P[b_t^i(s',s)]$. So the total computation load for n_0 coded bits is $n_0 \cdot (2^{k_0(\nu-1)} \cdot 2^{k_0} \cdot 2)$ MULs.

Combining all the above calculations, the total computation load for n_0 coded bits in one time cycle is:

$$\begin{aligned} L_{n_0}^{\text{MAP-DEC}} &= [(n_0-1)\cdot 2^{n_0}] + [2^{k_0} \cdot 2^{k_0(\nu-1)}] + [2^{k_0(\nu-1)} + 1] + [2^{k_0} \cdot 2^{k_0(\nu-1)}] \\ &\quad + [2^{k_0(\nu-1)}] + [2^{k_0(\nu-1)} \cdot 2^{k_0} \cdot 2 \cdot n_0] \\ &= (n_0-1)2^{n_0} + (2n_0 + 2 + 2^{1-k_0})2^{k_o \nu} + 1 \end{aligned} \tag{3.2}$$

For convenience in evaluating and comparing the overall complexity of different systems, the computation load in (3.2) is determined for each coded bit as

$$L^{\text{MAP-DEC}} \;=\; \frac{n_0 - 1}{n_0} 2^{n_0} + \frac{(2n_0 + 2 + 2^{1-k_0})}{n_o} 2^{k_o \nu} + \frac{1}{n_0} \tag{3.3}$$

3.1.2 Complexity of the MAP Demodulator

Recall that every symbol $c_k(l)$ carries J coded bits $[d_k(l,1), \cdots, d_k(l,j), \cdots, d_k(l,J)]^T$. The core of MAP demodulator involves the following three equations:

1. Equation (2.44):

$$P[C_i] \;=\; \prod_{j=1}^{J} P[d_k(l,j) = D(i,j)],\ i = 1, 2, \cdots, 2^J$$

   For each candidate symbol C_i, it takes $(J-1)$ MULs to calculate its own probability $P[C_i]$. So it takes $2^J \cdot (J-1)$ MULs to calculate all candidate symbols' probabilities for J coded bits.

2. Equation (2.40):

$$P[\hat{c}_k(l)|c_k(l) = C_i] = \frac{1}{\pi E\{\nu_k^2(l)\}} \cdot \exp\left(-\frac{\|\hat{c}_k(l) - \mu_k(l)C_i\|^2}{E\{\nu_k^2(l)\}}\right), \quad i = 1, 2, \cdots, M,\ C_i \in \Omega_C$$

   Given one candidate symbol C_i, it takes 4 MULs, plus one exponential operation, denoted by EXP, to compute the *a posterior* probability of the estimated symbol $P[\hat{c}_k(l)|c_k(l) = C_i]$. So for all the possible symbol candidates in Ω_C, it requires $2^J \cdot (4 \text{ MULs} + 1 \text{ EXP})$ to compute the *a posterior* probability of the estimated symbol.

3. Equation (2.43):

$$\begin{aligned}\Lambda_1[d_k(l,j)] \;&\triangleq\; \log \frac{P[d_k(l,j) = 1|\hat{c}_k(l)]}{P[d_k(l,j) = 0|\hat{c}_k(l)]} \\ &= \log \frac{\sum_{C_i \in C_j^+} P[\hat{c}_k(l)|c_k(l) = C_i]P[C_i]}{\sum_{C_i \in C_j^-} P[\hat{c}_k(l)|c_k(l) = C_i]P[C_i]}, \quad j = 1, 2, \cdots, J\end{aligned}$$

   For each coded bit carried by $c_k(l)$, it takes $(2^J + 1)$ MULs $+ 1$ LOG to compute the *a posterior* probability $\Lambda_1[d_k(l,j)]$. So in total, $J[(2^J + 1) \text{ MULs} + 1 \text{ LOG}]$ operations are needed to compute the *a posterior* probability of all J coded bits carried by symbol $c_k(l)$.

Then, the total computation load for the MAP demodulator per J coded bits (or per symbol) is given as

$$\begin{aligned} L_J^{\text{MAP-DEM}} &= [2^J \cdot (J-1)\text{MULs}] + [2^J \cdot (4\text{MULs} + 1\text{EXP})] + [J((2^J+1)\text{MULs} + 1\text{LOG})] \\ &= [2^J(2J+3) + J]\text{MULs} + 2^J\text{EXPs} + J\text{LOG} \end{aligned} \tag{3.4}$$

Again, normalizing by the number of coded bits carried by one symbol is useful and it gives

$$L^{\text{MAP-DEM}} = \left[\frac{(2J+3)}{J}2^J + 1\right]\text{MULs} + \frac{1}{J}2^J\text{EXPs} + 1\text{LOG} \tag{3.5}$$

3.1.3 Complexity of the MMSE Multiuser Detector

To aid the understanding in determining the computation load of the MMSE detector, the algorithm is briefly reviewed here. The MMSE detector processes the received signal block by block. Each block includes NK symbols, where N is the number of transmit symbols for one user during one block period and K is the number of the users. So, the computation load is first evaluated for NK symbols. It is then converted to the computation load per each transmitted symbol, and eventually per each coded bit.

1. Compute $\tilde{c}_k(l)$ and $E\{\|c_k(l)\|^2\}$ for all NK symbols ($k = 1, \cdots, K; l = 1, \cdots, N$) as

   $$\tilde{c}_k(l) \triangleq E[c_k(l)] = \sum_{C_i \in \Omega_C} C_i \, P[c_k(l) = C_i] \qquad \text{[This is Eqn. (2.16)]}$$

   $$E\{\|c_k(l)\|^2\} \triangleq \sum_{C_i \in \Omega_C} \|C_i\|^2 \, P[c_k(l) = C_i] \qquad \text{[This is Eqn. (2.17)]}$$

   Note that $P[c_k(l) = C_i]$ is already obtained in the MAP demodulator, and there is no need to calculate $P[c_k(l) = C_i]$ again in the MMSE detector. Since C_i is a complex symbol, the computation load for $C_i \, P[c_k(l) = C_i]$ is thus 2 MULs; and the computation load for $\|C_i\|^2 \, P[c_k(l) = C_i]$ is 1 MUL. Here it is not necessary to count the computation load for $\|C_i\|^2$ because it was already calculated at the very beginning before the iteration.

So for all NK symbols ($k = 1, \cdots, K; l = 1, \cdots, N$) in one block, the total computation load for $\tilde{c}_k(l)$ and $E\{\|c_k(l)\|^2\}$ is

$$L_1 = NK \cdot 2^J(2+1) = 3NK2^J \text{ (MULs)} \tag{3.6}$$

2. Compute $\tilde{\mathbf{r}}_k(l)$ for all NK symbols ($k = 1, \cdots, K; l = 1, \cdots, N$) as

$$\begin{aligned} \tilde{\mathbf{r}}_k(l) &\triangleq \mathbf{r} - \mathbf{H}\tilde{\mathbf{c}}_k(l) && \text{[This is Eqn. (2.20)]} \\ &= \mathbf{r} - \mathbf{H}[\tilde{\mathbf{c}} - \tilde{c}_k(l)\tilde{\mathbf{e}}_k(l)] \\ &= \underbrace{\mathbf{r} - \mathbf{H}\tilde{\mathbf{c}}}_{\text{Part 1}} - \underbrace{\tilde{c}_k(l)\mathbf{H}\tilde{\mathbf{e}}_k(l)}_{\text{Part 2}} \end{aligned}$$

Observe that Part 1 is common for all NK symbols and needs to be computed for only one time per block. Calculation of this part needs $MP \times NK$ complex multiplications, i.e., $4MPNK$ (MULs). The computation of Part 2 is more special and needs to be done separately for each symbol. For one symbol, Part 2 needs MP complex multiplications, i.e., $4MP$ (MULs). Therefore for all NK symbols in one block, the total computation load for $\tilde{\mathbf{r}}_k(l)$ is

$$L_2 = \underbrace{4MPNK}_{\text{Part 1}} + \underbrace{NK \times 4MP}_{\text{Part 2}} = 8MPNK \text{ (MULs)} \tag{3.7}$$

3. Compute $\hat{c}_k(l)$ for all NK symbols ($k = 1, \cdots, K; l = 1, \cdots, N$) as:

$$\hat{c}_k(l) = \tilde{\mathbf{e}}_k^H(l) \cdot \mathbf{H}^H \cdot [\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H + N_0\mathbf{I}]^{-1} \cdot \tilde{\mathbf{r}}_k(l) \cdot E\left\{||c_k(l)||^2\right\}$$

[This is Eqn. (2.31)]

The whole calculation can be decomposed into the following sub-steps:

(a) First we need to prepare $\mathbf{V}_k(l)$ according to (2.27). Because $E\{\|c_k(l)\|^2\}$ is already obtained in the first step, it only takes $2NK$ (MULs) to compute $\|\tilde{c}_k(l)\|^2, k = 1, \cdots, K; l = 1, \cdots, N$.

(b) In order to achieve an efficient calculation of $\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H$, define

$$\begin{aligned} \mathbf{V} \triangleq \text{diag}\{&E\{\|c_1(1)\|^2\} - \|\tilde{c}_1(1)\|^2, \cdots, \\ &E\{\|c_K(N)\|^2\} - \|\tilde{c}_K(N)\|^2\} \end{aligned} \tag{3.8}$$

Then

$$\begin{aligned}\mathbf{HV}_k(l)\mathbf{H}^H &= \mathbf{H}\left[\mathbf{V}+\mathrm{diag}\{0,0,\cdots,\|\tilde{c}_k(l)\|^2,0,0\}\right]\mathbf{H}^H \\ &= \underbrace{\mathbf{HVH}^H}_{\text{Part 1}}+\underbrace{[\mathbf{He}_k(l)]\|\tilde{c}_k(l)\|^2[\mathbf{He}_k(l)]^T}_{\text{Part 2}} \qquad (3.9)\end{aligned}$$

Part 1 is common for all NK symbols in one block, and it needs to be determined once per block. Because $\mathbf{V}$ is a real diagonal matrix and $\mathbf{H}$ is a complex matrix, $2\cdot NK\cdot MP$ (MULs) are needed to compute $\mathbf{VH}^H$ and $4\cdot NK\cdot(MP)^2$ (MULs) are needed to compute $\mathbf{HVH}^H$. So calculating Part 1 requires a total of $4NK(MP)^2+2NKMP$ (MULs). Next Part 2 needs to be computed separately for each individual symbol. For one symbol, $2MP$ (MULs) are required to compute $\|\tilde{c}_k(l)\|^2[\mathbf{He}_k(l)]^T$ and $4(MP)^2$ (MULs) are needed to compute $[\mathbf{He}_k(l)]\|\tilde{c}_k(l)\|^2[\mathbf{He}_k(l)]^T$. Therefore, to calculate Part 2 for all NK symbols in one block, a total of $NK[4(MP)^2+2MP]$ (MULs) computations are needed.

Combining the computations of Part 1 and Part 2, for all NK symbols in one block, it takes $8NK(MP)^2+4NKMP$ (MULs) to compute $\mathbf{HV}_k(l)\mathbf{H}^H$. Furthermore, given the symmetry property of the matrix, almost half of the computation load can be saved, so the computation load is counted as $4NK(MP)^2+2NKMP$ (MULs).

(c) Compute $\mathbf{T}_1 \triangleq [\mathbf{HV}_k(l)\mathbf{H}^H+N_0\mathbf{I}]^{-1}_{MP\times MP}$. This takes $4(MP)^3/3$ (MULs) [14] for one estimate $\hat{c}_k(l)$ to do the matrix inversion, so for all symbols in one block, it requires $\frac{4}{3}(MP)^3NK$ (MULs).

(d) Compute $\mathbf{T}_2 \triangleq \tilde{\mathbf{e}}_k^H(l)\cdot\mathbf{H}^H\cdot\mathbf{T}_1=\tilde{\mathbf{e}}_k^H(l)\cdot\mathbf{H}^H\cdot[\mathbf{HV}_k(l)\mathbf{H}^H+N_0\mathbf{I}]^{-1}$. Due to the effect of $\tilde{\mathbf{e}}_k^H(l)$, one only needs to compute one row of matrix $\{\mathbf{H}^H\cdot\mathbf{T}_1\}$ for one symbol. Therefore, it only takes $(MP)^2$ (MULs) to compute $\mathbf{T}_2$. So for all NK symbols, it requires $NK(MP)^2$ (MULs).

(e) Compute $\mathbf{T}_3 \triangleq \mathbf{T}_2\cdot\tilde{\mathbf{r}}_k(l)\cdot E\{||c_k(l)||^2\}$. For one symbol, it takes $4MP+2$ (MULs) to compute $\mathbf{T}_3$, which includes the inner product of two vectors and one complex weighting operation. So for all NK symbols in one block,

it requires $NK(4MP+2)$ (MULs).

In summary, in order to compute $\hat{c}_k(l)$ in (2.31), the computation load is

$$\begin{aligned} L_3 &= \underbrace{2NK}_{\text{(a)}} + \underbrace{4NK(MP)^2 + 2NKMP}_{\text{(b)}} + \underbrace{\frac{4}{3}(MP)^3 NK}_{\text{(c)}} + \underbrace{NK(MP)^2}_{\text{(d)}} \\ &\quad + \underbrace{NK(4MP+2)}_{\text{(e)}} \text{ (MULs)} \\ &= \frac{4}{3}NK(MP)^3 + 5NK(MP)^2 + 6NKMP + 4NK \text{ (MULs)} \quad (3.10) \end{aligned}$$

4. Compute $\mu_k(l)$ as follows:

$$\begin{aligned} \mu_k(l) &\triangleq \{\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H + \sigma^2\mathbf{I}]^{-1}\mathbf{H}\}_{kk} \cdot E\{\|c_k(l)\|^2\} \text{ [This is Eqn. (2.36)]} \\ &= \tilde{\mathbf{e}}_k^H(l) \cdot \mathbf{H}^H[\mathbf{H}\mathbf{V}_k(l)\mathbf{H}^H + \sigma^2\mathbf{I}]^{-1}\mathbf{H} \cdot \tilde{\mathbf{e}}_k(l) \cdot E\{\|c_k(l)\|^2\} \\ &= \mathbf{T}_2 \cdot \{\mathbf{H} \cdot \tilde{\mathbf{e}}_k(l)\} \cdot E\{\|c_k(l)\|^2\} \quad (3.11) \end{aligned}$$

Since $\mathbf{T}_2$ is already available from the previous calculation, it only takes $2MP$ (MULs) to calculate the inner product of $\mathbf{T}_2 \cdot \{\mathbf{H} \cdot \tilde{\mathbf{e}}_k(l)\}$ and 1 (MUL) to weight the result by $E\{\|c_k(l)\|^2\}$. So the total computation load for all NK symbols is:

$$L_4 = NK(2MP+1) \text{ (MULs)} \quad (3.12)$$

5. Compute $E\{\nu_k^2(l)\}$ as:

$$E\{\nu_k^2(l)\} = [\mu_k(l) - \mu_k^2(l)]E\{\|c_k(l)\|^2\} \text{ [This is Eqn. (2.39)]}$$

For one symbol, it takes 2 (MULs) to compute $E\{\nu_k^2(l)\}$. Thus the total computation load for all NK symbols is:

$$L_5 = 2NK \text{ (MULs)} \quad (3.13)$$

Combining all the computation loads as given in (3.6), (3.7) (3.10), (3.12), and (3.13), the total complexity of the MMSE multiuser detector per NK symbols is:

$$\begin{aligned} L_{\text{total}}^{\text{MMSE-MUD}} &= \underbrace{3NK2^J}_{L_1} + \underbrace{8MPNK}_{L_2} + \\ &\quad \underbrace{\frac{4}{3}NK(MP)^3 + 5NK(MP)^2 + 6NKMP + 4NK}_{L_3} \\ &\quad + \underbrace{NK(2MP+1)}_{L_4} + \underbrace{2NK}_{L_5} \\ &= \frac{4}{3}NK(MP)^3 + 5NK(MP)^2 + 16NKMP + NK[3 \times 2^J + 6] \text{ (MULs)} \end{aligned}$$

Finally, the computation load per one coded bit is expressed as

$$\begin{aligned} L^{\text{MMSE-MUD}} &= \frac{L_{\text{total}}^{\text{MMSE-MUD}}}{NK \cdot J} \\ &= \frac{4}{3J}(MP)^3 + \frac{5}{J}(MP)^2 + \frac{16}{J}MP + \frac{[3 \cdot 2^J + 6]}{J}(\text{MULs}) \quad (3.14) \end{aligned}$$

3.2 Efficiency Analysis of the Iterative Receiver

Typically, the bit-error rate (BER) curves of an iterative receiver can be divided into three regions [15]: 1) the region of low E_b/N_0 with negligible iterative BER reduction, 2) the turbo cliff region (also referred to as "waterfall"-region) with persistent iterative BER reduction over many iterations, and 3) the BER floor region for moderate to high E_b/N_0 in which a rather low BER floor region can be reached after a few number of iterations.

While analytical bounding techniques have been successfully applied to the asymptotic performance for moderate to high E_b/N_0, they are not good enough to analyze the efficiency of the iterative process (i.e., how fast the iterative receiver can approach the asymptotic BER performance), especially for the turbo-cliff region. In [16], a density evolution algorithm is proposed to investigate the convergence behavior of iterative decoding. The algorithm can calculate convergence thresholds for low-density parity-check (LDPC) codes over an AWGN channel by investigating the probability density functions (PDF) of the communicated information within the iterative decod-

ing algorithm. In [17], the convergence of iterative decoders is studied based on SNR measures. In [15], the author proposes extrinsic information transfer characteristics based on mutual information to describe the flow of extrinsic information through the soft-input/soft-output constituent decoders. The technique proves to be particularly useful in the analysis of the region of low E_b/N_0.

Specifically, the exchange of extrinsic information between the constituent decoders is visualized by a decoding trajectory in the extrinsic information transfer chart (EXIT chart). By observing only single parameters of the PDFs and assuming those PDFs are Gaussian, EXIT chart has also been applied successfully to various concatenated systems, including both parallel and serially concatenated codes [18] [19]. A comparative study observing six different parameters also revealed that mutual information is one of the most accurate and robust parameters [20].

This thesis also chooses to use mutual information and EXIT chart to investigate the convergence behavior of the iterative processing employed by the receiver in multiuser STBC systems. To this end, the following subsection reviews the process of the conventional iterative decoding scheme. It then presents how to construct the corresponding EXIT charts to analyze the efficiency of the iterative receiver.

3.2.1 Iterative Decoding of Multiuser STBC Systems

For convenience, Fig. 3.1 displays again the block diagram of the conventional iterative receiver for a multiuser STBC system.

The coded bit $d_k(l, j)$ can be modeled as binary random variable X_k. At the transmitter side, the bits $d_k(l, j)$ are assumed to be equally likely, i.e., $P(d_k(j) = 1) = P(d_k(j) = 0) = 1/2$. The LLRs $\{\Lambda_1[d_1(k, j)]\}$, $\{\lambda_1[d_1(k, j)]\}$, and $\{\lambda_1^{\pi}[d_1(k, j)]\}$ can be modeled as random variables $D_{1,k}$, $E_{1,k}$, and $A_{2,k}$, respectively; The LLRs $\{\Lambda_2[d_k(l, j)]\}$, $\{\lambda_2[d_k(l, j)]\}$, and $\{\lambda_2^{\pi}[d_k(l, j)]\}$ are modeled as random variables $D_{2,k}$, $E_{2,k}$, and $A_{1,k}$, respectively. Since each user's decoder works independently and in the same way, the subscript k (the user index) is dropped in the following sections to simplify the notations.

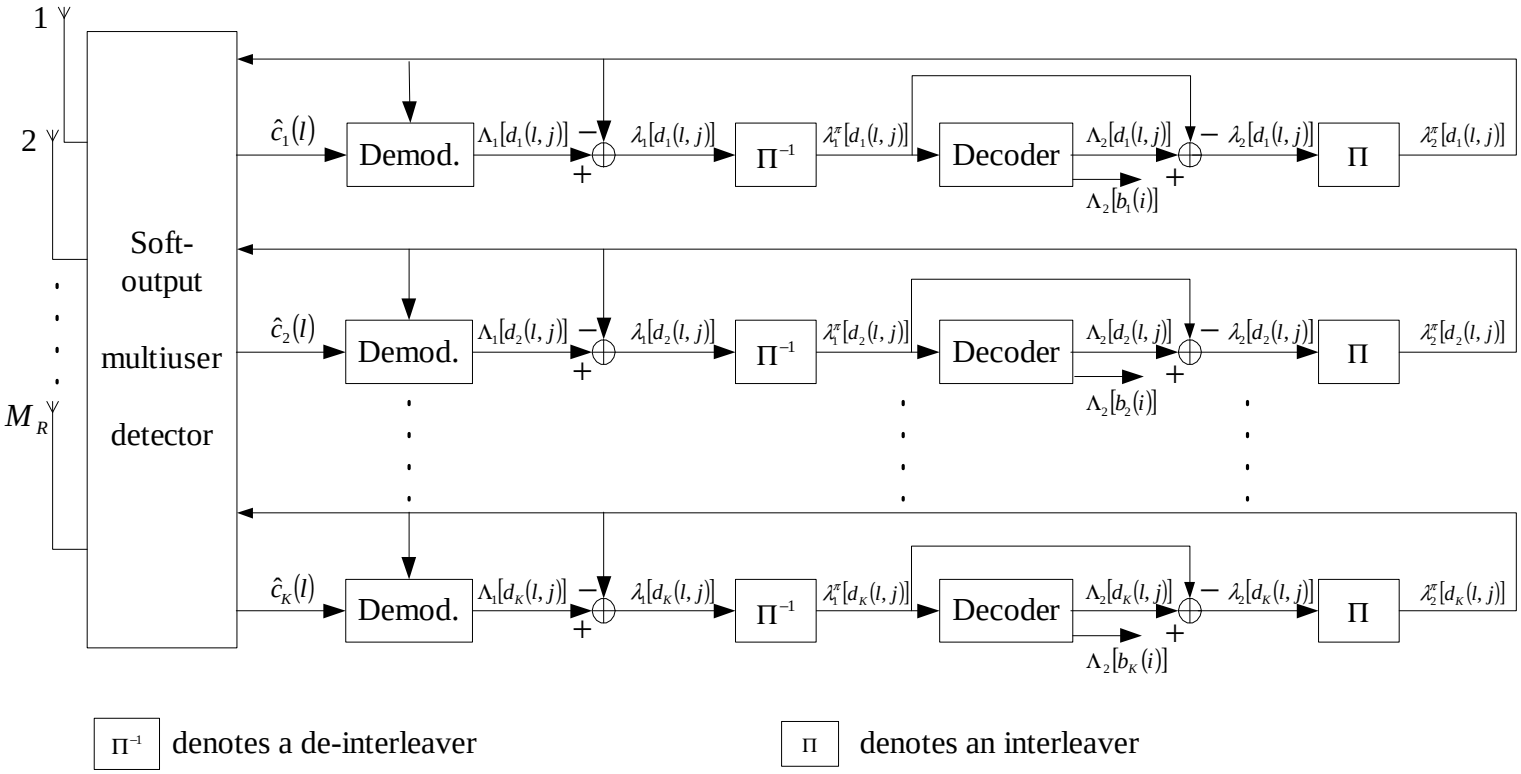


Figure 3.1 Conventional iterative receiver for a multiuser STBC system (reproduced from Fig. 2.4).

At the receiver the signal is iteratively decoded by exchanging the soft information between the MUD/DEMO and the decoders as shown in Figure 3.2.

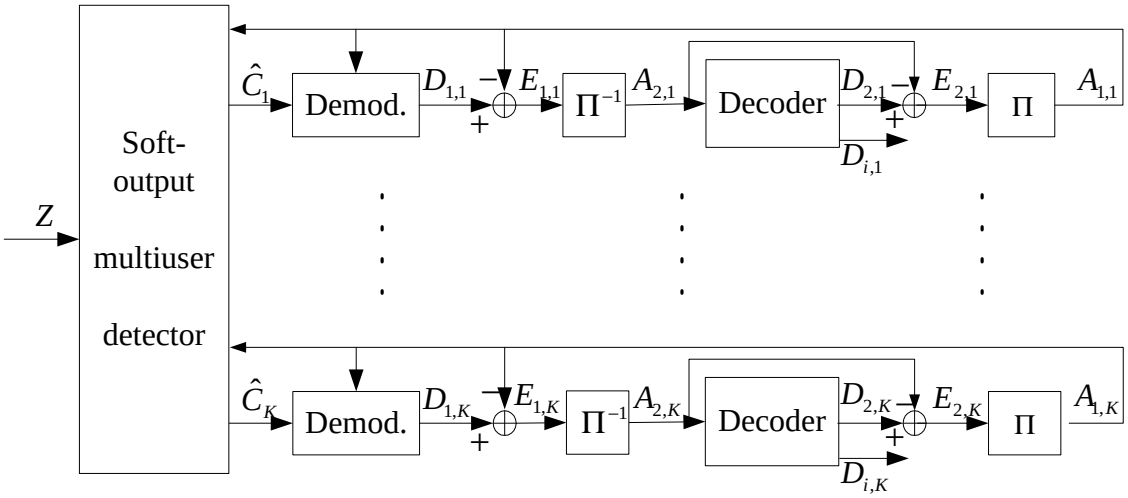


Figure 3.2 Extrinsic information flow of the receiver with iterative decoding.

The random variables $(D_1, A_1, E_1, D_2, A_2, E_2)$ are described with the conditional PDFs $p(l|X = \pm 1)$, which changes with the iterations. Analyzing these PDFs allows one to predict the behavior of the decoding algorithm, i.e., the efficiency of the iterative scheme. The following two useful observations are obtained by simulation results in [15]:

1. For large interleavers, the LLR values of all these random variables remain fairly uncorrelated from the respective channel observation Z over many iterations.

2. The probability density functions of all the LLR values are almost Gaussian distributed.

3.2.2 Mutual Information

In general, the mutual information between two random variables is a measure of information provided by one variable about the other random variable [21]. As mentioned earlier, mutual information between the coded bits and random variables D_1, A_1, E_1, D_2, A_2, and E_2 (all are the LLR values of the coded bits) is an accurate

and robust parameter to observe the behavior of the iterative receiver based on the corresponding conditional PDFs.

Without loss of generality, take A_1 as an example to show how the mutual information is defined and used. Observations 1 and 2 in the previous section suggest that the *a priori* input A_1 to the MUD/DEMO can be modeled as an zero-mean independent Gaussian random variable n_{A_1} with variance $\sigma_{A_1}^2$ in conjunction with the coded bits $x \in \{\pm 1\}$ [15]:

$$A_1 = \mu_{A_1} \cdot x + n_{A_1} \tag{3.15}$$

Since A_1 is supposed to be an LLR value based on the Gaussian distribution, it can be shown that μ_{A_1} must fulfill $\mu_{A_1} = \sigma_{A_1}^2/2$ [22]. Thus the conditional PDF of A_1 is

$$p_{A_1}(\xi|X=x) = \frac{e^{-\frac{\left(\xi - \frac{\sigma_{A_1}^2}{2}x\right)^2}{2\sigma_{A_1}^2}}}{\sqrt{2\pi}\sigma_{A_1}} \tag{3.16}$$

To measure the information content of the *a priori* knowledge, the mutual information $I_{A_1} = I(X; A_1)$ between the coded bit X and the LLR value A_1 is used and can be computed as follows (a more detailed explanation of mutual information is provided in Appendix B):

$$\begin{aligned} I_{A_1} &= \frac{1}{2} \cdot \sum_{x=-1,1} \int_{-\infty}^{+\infty} p_{A_1}(\xi|X=x) \\ &\qquad \times \log_2 \frac{2 \cdot p_{A_1}(\xi|X=x)}{p_{A_1}(\xi|X=-1) + p_{A_1}(\xi|X=1)} d\xi \end{aligned} \tag{3.17}$$

Note that $0 \leq I_{A_1} \leq 1$. In [23], $I_{A_1}(\sigma_{A_1})$ is given as

$$I_{A_1}(\sigma_{A_1}) = 1 - \int_{-\infty}^{+\infty} \frac{e^{-\frac{(\xi - \frac{\sigma_{A_1}^2}{2})^2}{2\sigma_{A_1}^2}}}{\sqrt{2\pi}\sigma_{A_1}} \cdot \log_2[1 + e^{-\xi}] d\xi \tag{3.18}$$

The function $I_{A_1}(\sigma_{A_1})$ cannot be expressed in a closed-form. It is monotonically increasing and thus reversible with

$$\lim_{\sigma_{A_1} \to 0} I_{A_1}(\sigma_{A_1}) = 0, \lim_{\sigma_{A_1} \to \infty} I_{A_1}(\sigma_{A_1}) = 1, \sigma_{A_1} > 0 \tag{3.19}$$

The following shows in a different way how $I_{A_1}(\sigma_{A_1})$ can be effectively computed numerically. First define $I_{A_1}^{(1)}$ and $I_{A_1}^{(-1)}$ as

$$I_{A_1}^{(1)} = \int_{-\infty}^{+\infty} p_{A_1}(\xi|X=1) \times \log_2 \frac{2 \cdot p_{A_1}(\xi|X=1)}{p_{A_1}(\xi|X=-1) + p_{A_1}(\xi|X=1)} d\xi \quad (3.20)$$

$$I_{A_1}^{(-1)} = \int_{-\infty}^{+\infty} p_{A_1}(\xi|X=-1) \times \log_2 \frac{2 \cdot_{A_1}(\xi|X=-1)}{p_{A_1}(\xi|X=-1) + p_{A_1}(\xi|X=1)} d\xi \quad (3.21)$$

Obviously,

$$I_{A_1} = \frac{1}{2}\left[I_{A_1}^{(1)} + I_{A_1}^{(-1)}\right] \quad (3.22)$$

From (3.16), one can obtain

$$\frac{p_{A_1}(\xi|X=1)}{p_{A_1}(\xi|X=-1)} = e^{\xi} \quad \text{or} \quad \frac{p_{A_1}(\xi|X=-1)}{p_{A_1}(\xi|X=1)} = e^{-\xi} \quad (3.23)$$

Taking the logrithm of both sides of (3.23) gives

$$\xi = \ln\left\{\frac{p_{A_1}(\xi|X=1)}{p_{A_1}(\xi|X=-1)}\right\} \quad (3.24)$$

The above expression shows that ξ is a valid value of an LLR variable. Substituting (3.23) into (3.20) and (3.21) produces

$$\begin{aligned} I_{A_1}^{(1)} &= \int_{-\infty}^{+\infty} p_{A_1}(\xi|X=1) \times \log_2 \frac{2}{1+\frac{p_{A_1}(\xi|X=-1)}{p_{A_1}(\xi|X=1)}} d\xi \\ &= 1 - \int_{-\infty}^{+\infty} p_{A_1}(\xi|X=1) \times \log_2[1+e^{-\xi}] d\xi \\ &= E_{X=1}\{1 - \log_2[1+e^{-\xi}]\} \end{aligned} \quad (3.25)$$

and

$$\begin{aligned} I_{A_1}^{(-1)} &= \int_{-\infty}^{+\infty} p_{A_1}(\xi|X=-1) \times \log_2 \frac{2}{1+\frac{p_{A_1}(\xi|X=1)}{p_{A_1}(\xi|X=-1)}} d\xi \\ &= 1 - \int_{-\infty}^{+\infty} p_{A_1}(\xi|X=-1) \times \log_2[1+e^{\xi}] d\xi \\ &= E_{X=-1}\{1 - \log_2[1+e^{\xi}]\} \end{aligned} \quad (3.26)$$

It follows that

$$\begin{aligned} I_{A_1} &= \frac{1}{2}\left[I_{A_1}^{(1)} + I_{A_1}^{(-1)}\right] \\ &= \frac{1}{2}E_{X=1}\{1 - \log_2[1+e^{-\xi}]\} + \frac{1}{2}E_{X=-1}\{1 - \log_2[1+e^{\xi}]\} \end{aligned} \quad (3.27)$$

These expectations over the PDFs $p_{A_1}(\xi|X=1)$ and $p_{A_1}(\xi|X=-1)$ can be closely approximated with an arbitrary accuracy by the time average [24]. That is,

$$I_{A_1} = \frac{1}{2}\left\{\underbrace{\frac{1}{L_1}\sum_{1}^{L_1}[1-\log_2(1+e^{-\lambda_{A_1}})]}_{X=1} + \underbrace{\frac{1}{L_2}\sum_{1}^{L_2}[1-\log_2(1+e^{\lambda_{A_1}})]}_{X=-1}\right\} \quad (3.28)$$

where L ($L = L_{X=1} + L_{X=-1}$) is the total number of bits in one simulation block. Here $L_{X=1}$ is the number of bits 1 and $L_{X=-1}$ is the number of bits -1. Because the transmitted bits are assumed to equally likely, (3.28) can be further simplified by setting $L_{X=1} = L_{X=-1} = L/2$ as:

$$\begin{aligned} I_{A_1} &= \frac{1}{2}\left\{\underbrace{\frac{1}{L/2}\sum_{n=1}^{L/2}[1-\log_2(1+e^{-\lambda_{A_1}})]}_{X=1} + \underbrace{\frac{1}{L/2}\sum_{n=1}^{L/2}[1-\log_2(1+e^{\lambda_{A_1}})]}_{X=-1}\right\} \\ &= \frac{1}{L}\left\{\underbrace{\sum_{n=1}^{L/2}[1-\log_2(1+e^{-\lambda_{A_1}})]}_{X=1} + \underbrace{\sum_{n=1}^{L/2}[1-\log_2(1+e^{\lambda_{A_1}})]}_{X=-1}\right\} \\ &= 1-\frac{1}{L}\sum_{n=1}^{L}\log_2(1+e^{-x\lambda_{A_1}}) \end{aligned} \quad (3.29)$$

where λ_{A_1} is the LLR value of the coded bit that is fed back to the MUD/DEMO, defined as

$$\lambda_{A_1} = \ln\frac{P(x=1)}{P(x=-1)} \quad (3.30)$$

Similarly, one can efficiently compute I_{E_1}, I_{A_2}, and I_{E_2} as

$$I_{E_1} = 1-\frac{1}{L}\sum_{n=1}^{L}\log_2(1+e^{-x\lambda_{E_1}}) \quad (3.31)$$

$$I_{A_2} = 1-\frac{1}{L}\sum_{n=1}^{L}\log_2(1+e^{-x\lambda_{A_2}}) \quad (3.32)$$

$$I_{E_2} = 1-\frac{1}{L}\sum_{n=1}^{L}\log_2(1+e^{-x\lambda_{E_2}}) \quad (3.33)$$

3.2.3 Transfer Characteristic of the MUD/DEMO

The MUD takes as its input the *a priori* knowledge $A_{1,k}$ $(k = 1, \cdots, K)$ of K users' coded bits and the channel observation Z. It outputs the estimates of the transmitted symbols $\hat{C}_k$. Based on the estimates $\hat{C}_k$ and the same *a priori* knowledge $A_{1,k}$, each user's demodulator outputs its own extrinsic information $E_{1,k}$ independently. To observe how the extrinsic information flows among three modules (MUD, K modulators, and K decoders), we first treat the MUD and all the K demodulators as one module to analyze its transfer characteristic. This is illustrated in Figure 3.3.

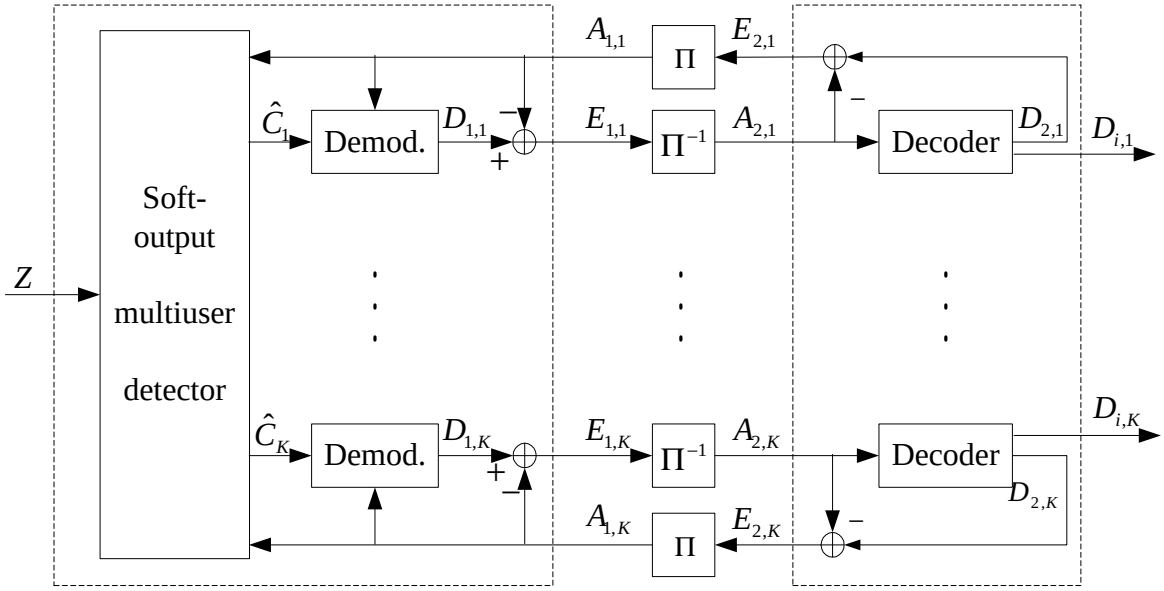


Figure 3.3 Extrinsic information flow of the conventional receiver.

Viewing $I_{E_{1,k}}$ as a function of $I_{A_{1,k}}$ and E_b/N_0, the extrinsic information transfer characteristic is defined as

$$I_{E_1} = T_1(I_{A_1}, E_b/N_0) \tag{3.34}$$

where I_{E_1} and I_{A_1} are defined as the average values of $I_{E_{1,k}}$ and $I_{A_{1,k}}$ for all users, respectively. To calculate the characteristic $T_1(I_{A_1}, E_b/N_0)$ for a desired $(I_{A_1}, E_b/N_0)$ input combination, the independent Gaussian random variable of (3.15) is applied as the *a priori* input to the MUD/DEMO of interest. Note that a certain value of I_{A_1}

is obtained by approximately choosing the parameter σ_{A_1} with $\sigma_{A_1} = I_{A_1}^{-1}(\sigma_{A_1})$. For this, the characteristics are most conveniently determined by means of Monte Carlo simulation.

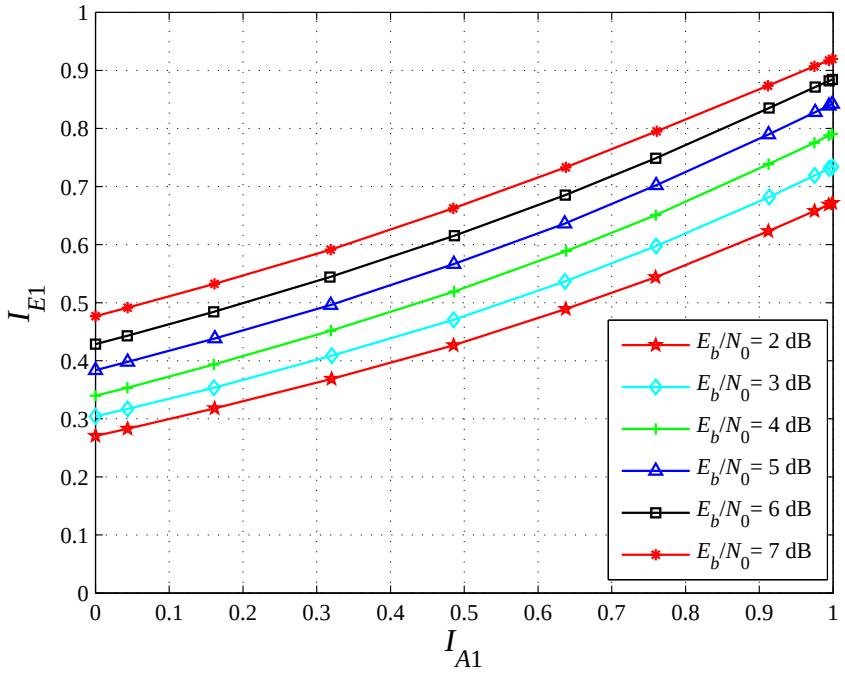


Figure 3.4 Extrinsic information transfer characteristic of the MUD together with demodulators for 8QAM-SSP mapping.

For illustration, various transfer characteristics obtained for 8QAM with Semi-Set Partitioning (SSP) mapping (see Fig. 5.4) are plotted in Fig. 3.4. The *a priori* input I_{A_1} is on the abscissa, the extrinsic output I_{E_1} is on the ordinate. The E_b/N_0-values serve as parameters of the curves. It should be pointed out that $I_{A_1}(0)$ means that the MUD/DEMO works with no *a priori* information about the coded bits. Similarly, $I_{A_1}(1)$ means that the MUD/DEMO works with perfect *a priori* information of the coded bits.

3.2.4 Transfer Characteristic of the Channel Decoder

The extrinsic transfer characteristic of the channel decoder is

$$I_{E_{2,k}} = T_2(I_{A_{2,k}}), \quad k = 1, \cdots, K \tag{3.35}$$

which describes the input/output relationship between the channel decoder's input $I_{A_{2,k}}$ and its extrinsic output $I_{E_{2,k}}$. Because all users employ the same convolutional code and decoder, we only need to investigate one of the channel decoders. Omitting the subscripts k simplifies (3.35) to

$$I_{E_2} = T_2(I_{A_2}) \tag{3.36}$$

Note that the characteristic $T_2(\cdot)$ does not depend on the E_b/N_0-value. By assuming that A_2 and E_2 are Gaussian distributed, I_{A_2} and I_{E_2} can be computed according to (3.32) and (3.33), respectively.

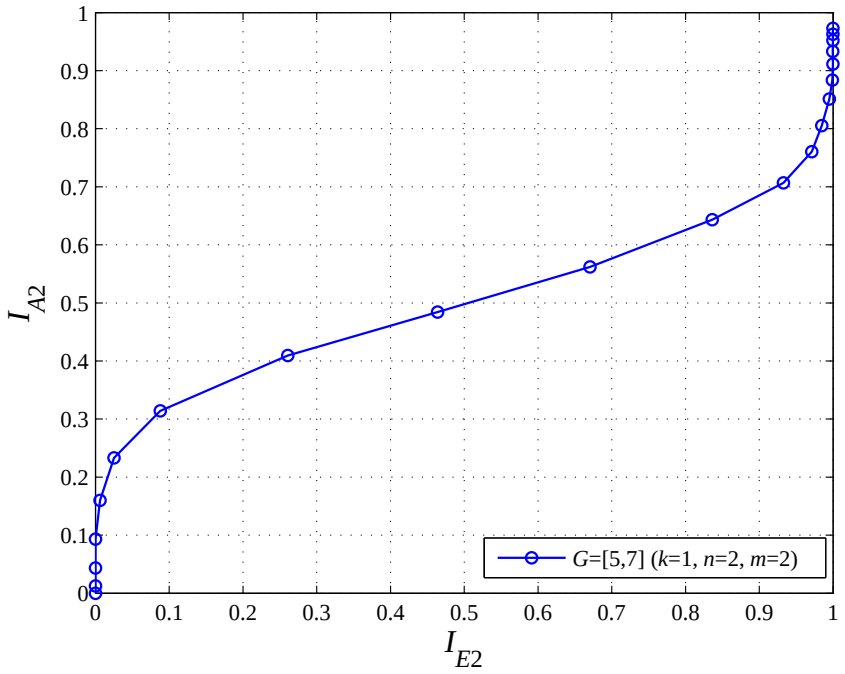


Figure 3.5 Extrinsic information transfer characteristic of the channel decoder.

As an example, Fig. 3.5 shows the extrinsic transfer characteristic of the channel decoder for a convolutional code with generator polynomial $G = (5, 7)$. Note that

the axes are swapped. The input I_{A_2} is on the ordinate and the output I_{E_2} is on the abscissa.

3.2.5 EXIT Charts for Multiuser STBC Systems

Connected through interleavers and de-interleavers, the extrinsic output E_1 of the MUD/DEMO becomes the *a priori* input A_2 to the decoder, and the extrinsic output E_2 of the decoder becomes the *a priori* input A_1 to the MUD/DEMO. Interleaving does not change the mutual information, so $I_{A_2} = I_{E_1}$, $I_{A_1} = I_{E_2}$. This exchange of extrinsic information is visualized in the extrinsic information transfer chart (EXIT chart) by plotting the MUD/DEMO and the decoder characteristics on a single diagram. Here I_{A_1} and I_{E_2} are on the abscissa, I_{A_2} and I_{E_1} are on the ordinate. The corresponding EXIT charts are shown in Figure 3.6 with E_b/N_0 as a parameter. Note that the characteristics of MUD/DEMO and decoders in EXIT charts are obtained separately and not in conjunction with any particular system simulation. It can be seen that as the E_b/N_0 value increases, the characteristic of MUD/DEMO raises. When E_b/N_0 is higher than some threshold (4dB as in Figure 3.6), a narrow tunnel is opened between the characteristic of the MUD/DEMO and that of the decoder, which allows for convergence of iterative decoding toward low BER.

This observation agrees well with the BER curves shown in Fig. 3.7 with four iterations. For $E_b/N_0 = 3$dB, iterative decoding has little improvement on the system performance. This is because in EXIT chart, two characteristic curves of the MUD/DEMO and the decoder are stuck together in a very early stage. When $E_b/N_0 = 4$dB, iterative decoding starts to work and the system performance improves along with the decoding iterations in a slow rate. This is in agreement with the in EXIT chart, where the two characteristic curves of the MUD/DEMO and the decoder open a narrow tunnel. When $E_b/N_0 = 6$dB (which is significantly bigger than the threshold 4dB), the system performance improves very quickly with decoding iterations. This is also evidenced from the EXIT chart as the two characteristic curves open a wide enough tunnel.

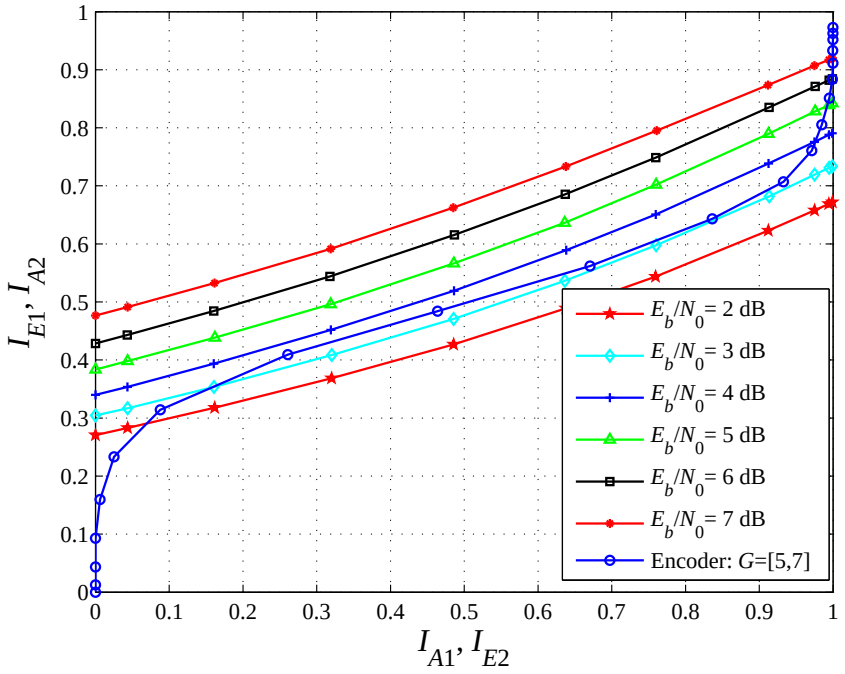


Figure 3.6 EXIT charts of the iterative receiver with E_b/N_0 as a parameter.

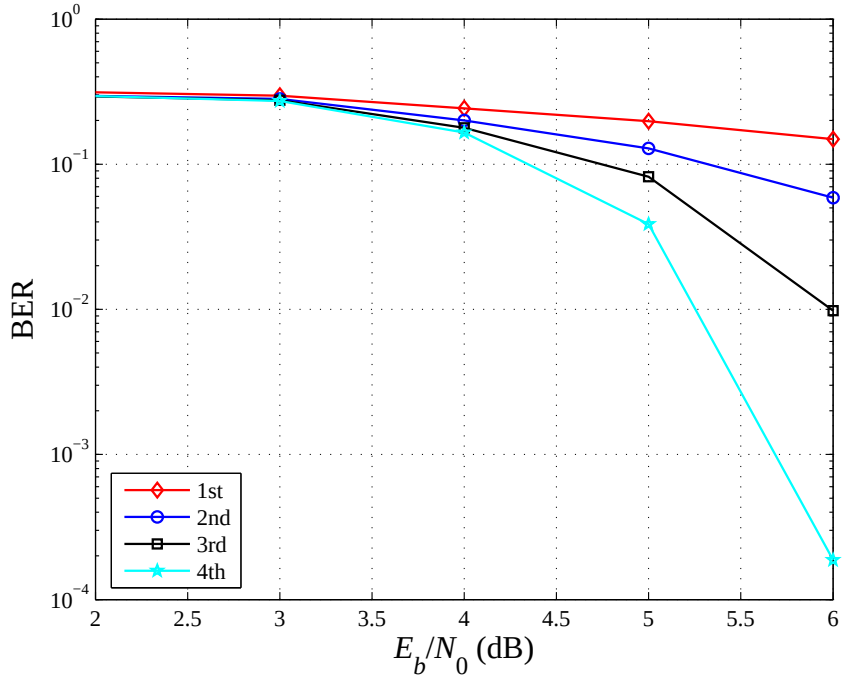


Figure 3.7 BER performance of the conventional iterative receiver.

The above example shows that the true behavior of the iterative multiuser receiver with iterative decoding can be described approximately by the individual transfer characteristics of the MUD/DEMO and decoders, provided that the independence and Gaussian assumptions of the extrinsic information (or *a priori* information) for (3.15) hold over many iterations, which is to a great extent guaranteed by the large interleaver. Besides the effect of the large interleaver, it also owns to the robustness of the mutual information measure, which overcomes some distortion of the *a priori* information distribution from the typical Gaussian distribution. For further demonstration and verification, the trajectory of iterative decoding is evaluated by means of Monte Carlo simulation under different E_b/N_0, and shown in Figs. 3.8, 3.9, and 3.10, respectively. The interleaver length used here is 12,000 bits/frame.

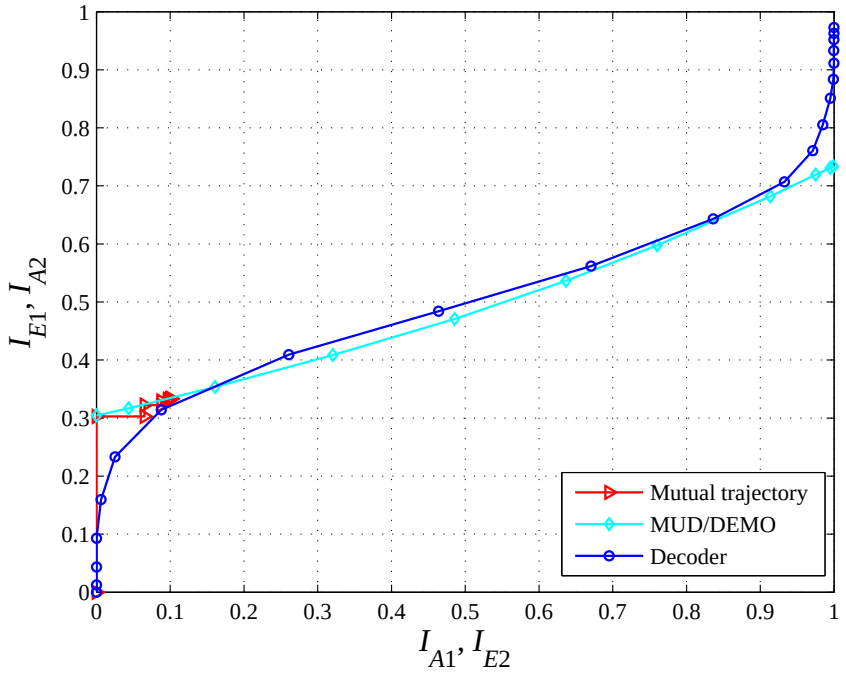


Figure 3.8 EXIT charts with iterative decoding trajectory at $E_b/N_0 = 3.0$dB.

The simulated trajectories match the individual characteristics of the MUD/DEMO and the decoders very well. It confirms the observations from EXIT charts that the iterative decoding is stuck very earlier for $E_b/N_0 = 3$dB (Figure 3.8) and can pass

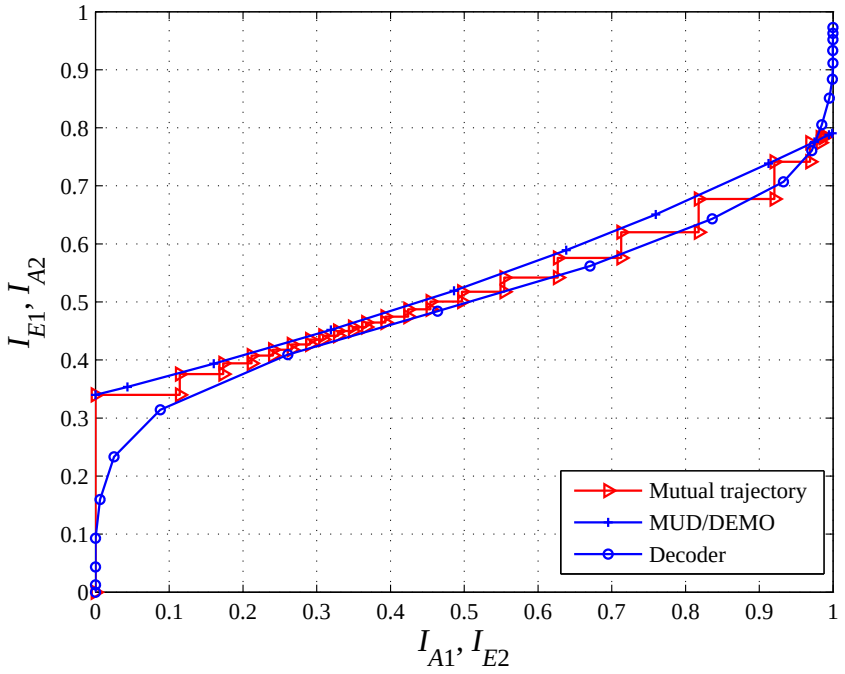


Figure 3.9 EXIT charts with iterative decoding trajectory at $E_b/N_0 = 4.0$dB.

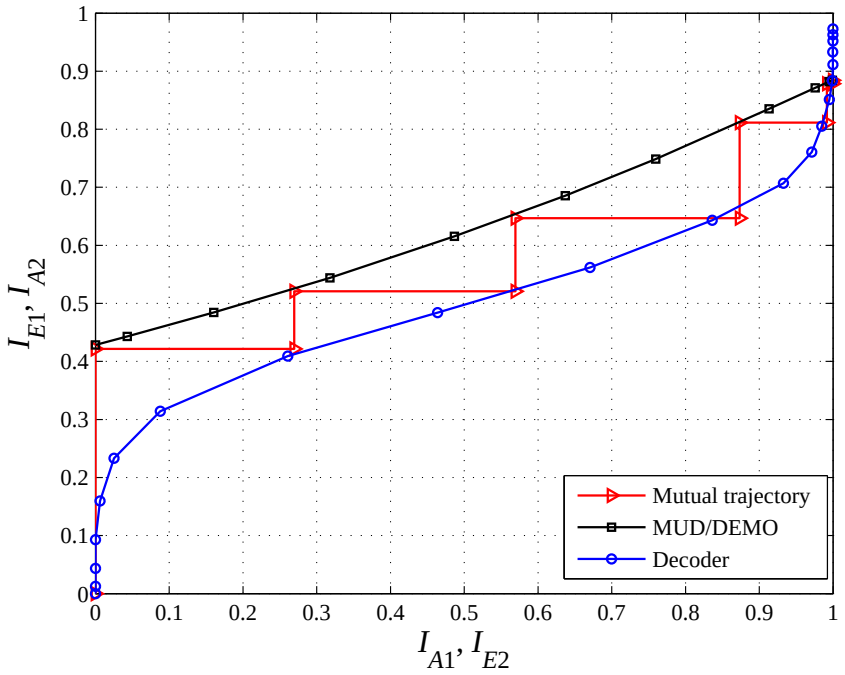


Figure 3.10 EXIT charts with iterative decoding trajectory at $E_b/N_0 = 6.0$dB.

through a narrow tunnel for $E_b/N_0 = 4$dB with enough iterations (Figure 3.9). And for $E_b/N_0 = 6$dB iterative decoding works effectively and approach the asymptotic performance with a few iterations (Figure 3.10). Figure 3.11 shows that the BER curves match well with the corresponding iterative decoding trajectory. In particular, the figures shows that the BER improves slowly when the iterative decoding trajectory tries to pass through the narrow tunnel. Then the BER curve shows a water-fall region right after the iterative decoding trajectory passes the narrow tunnel.

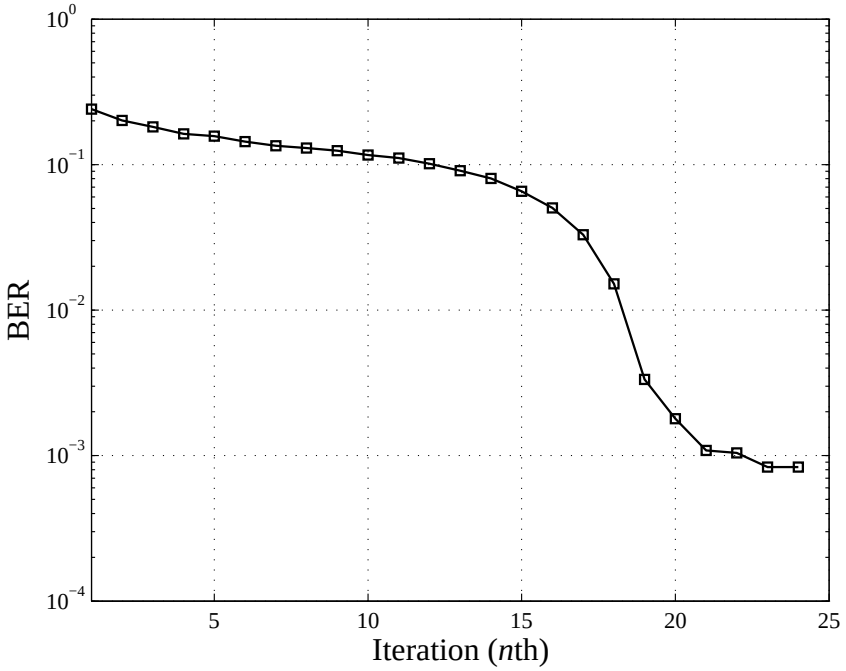


Figure 3.11 BER with iterations at $E_b/N_0 = 4.0$dB.

Finally, it should be pointed out that the Gaussian and independence assumptions on the *a priori* inputs $A_{1,k}$ ($E_{2,k}$) and $A_{2,k}$ ($E_{1,k}$) are only imposed to calculate the individual characteristics of the MUD/DEMO and the decoders. The decoding trajectory is a simulation result purely based on measurement of the extrinsic information at the outputs of the MUD/DEMO and the decoders during system simulation.

Given the effectiveness and simplicity of the EXIT chart analysis in predicting the

convergence behavior of the iterative, this technique will be used extensively in the following chapters to compare and study different decoding schemes. This technique helps to avoid time consuming implementation of the whole system by Monte Carlo simulation.

4. Sigma Mapping and its Application to STBC Systems

As seen in Chapter 3, the complexity of iterative receiver in a STBC system is mainly determined by the complexity of the soft-output MUD, the MAP demodulator and the MAP convolutional decoder. The MAP decoder is a fairly standard block and a low-complexity MUD for M-QAM is already discussed in Chapter 3. In this chapter, more attention shall be placed on MAP demodulator employed by each user.

For single-user communications over a flat-fading channel, sigma mapping has been shown to be extremely useful in bit-interleaved coded modulation with iterative decoding (BICM-ID) to reduce the demodulator'complexity while maintaining the excellent error performance of the system [25]. The basic idea behind sigma mapping is to relate the vectors of binary coded bits to the transmitted symbols in a linear manner, which then ease the demodulation process at the receiver. This type of mapping scheme was originally proposed as a capacity-approaching mapping method for a multilevel coding scheme [26]. In particular, a detailed investigation was carried out for M-PAM constellations when antipodal signals are adopted for modulation from coded bits to transmitted symbols [27].

Motivated by these results, it is natural and interesting to apply the sigma mapping for multiuser STBC systems. By exploiting the advantage of the linear relationship between the coded bits and the transmitted symbol, a suboptimal soft-output minimum mean-square error (MMSE) demodulator is developed. Furthermore the linearity property of sigma mapping makes it possible to integrate an individual's demodulators into the MUD. Then a low-complexity iterative receiver will be presented

to improve the convergence of the iterative processing by working on the bit-level interference cancellation.

4.1 Definition of Sigma Mapping

The basic idea of sigma mapping is to relate the vectors of coded bits to the transmitted complex symbols in a linear pattern [25]. Precisely, sigma mapping can be defined as

$$x = \underline{V} \cdot (2\underline{b} - \underline{1}) \tag{4.1}$$

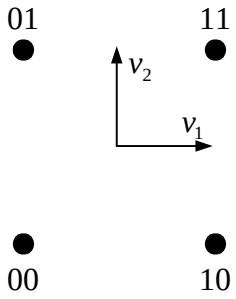
where x is the modulated symbol to be transmitted over the channel, $\underline{b}$ is a vector of J coded bits (0 or 1) and $\underline{1}$ is a all-one vector (of J elements). The vector $\underline{V} = [v_1, v_2, \cdots, v_J]$ contains J complex numbers, which are named as basis vectors. The basis vectors are chosen to satisfy

$$\sum_{j=1}^{J} \|v_i\|^2 = E_s \tag{4.2}$$

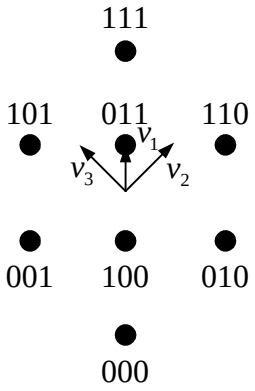
where E_s is the average symbol energy.

For a fixed set of basis vectors, there are $M = 2^J$ different channel symbols, which create an M-ary sigma constellation, denoted by $\Omega_{\sum}$. Obviously the shape of $\Omega_{\sum}$ depends on the basis vectors $\{v_j, 1 \leq j \leq J\}$. It can be shown that any M-QAM ($M = 2^J$) constellation can be considered as an $\Omega_{\sum}$ constellation by selecting a proper set of basis vectors [25]. As examples, Fig. 4.1 presents the sigma mappings for QPSK, 8-QAM, and 16-QAM constellations, where the corresponding basis vectors are also indicated in the figures. Note that sigma mapping is exactly the Gray mapping for QPSK constellation. The technique of sigma mapping can also be extended to non-traditional constellations (i.e., with arbitrary shapes) with a more flexibility in the design of the basis vectors $\{v_j, 1 \leq j \leq J\}$.

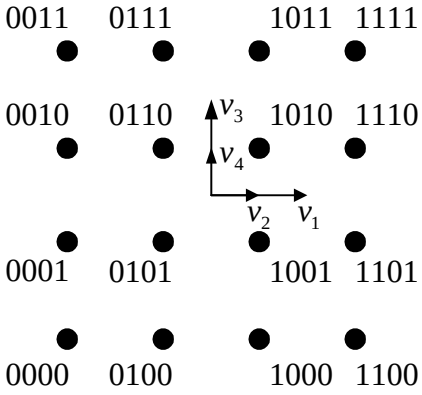
Fig. 4.2 illustrates how the sigma mapping is embedded in a coded modulation system. A binary information sequence $\underline{u}$ is encoded by a convolutional encoder. The resulting coded sequence $\underline{d}$ is then fed into a random interleaver. The randomly-interleaved version $\underline{\tilde{d}}$ of $\underline{d}$ is converted by a serial-to-parallel converter (S/P) into J



(a) Sigma mapping for QPSK.



(b) Sigma mapping for 8-QAM.



(c) Sigma Mapping for 16-QAM.

Notes:

(a): $v_1 = \exp(j0),\ v_2 = \exp\left(j\frac{\pi}{2}\right)$

(b): $v_1 = \sqrt{0.6}\exp\left(j\frac{\pi}{2}\right),\ v_2 = \sqrt{1.2}\exp\left(j\frac{\pi}{4}\right),$

$v_3 = \sqrt{1.2}\exp\left(j\frac{3\pi}{4}\right)$

(c): $v_1 = \sqrt{1.6}\exp(j0),\quad v_2 = \sqrt{0.4}\exp(j0),$

$v_3 = \sqrt{1.6}\exp\left(j\frac{\pi}{2}\right),\ v_4 = \sqrt{0.4}\exp\left(j\frac{\pi}{2}\right)$

Figure 4.1 Sigma mapping for various M-QAM constellations.

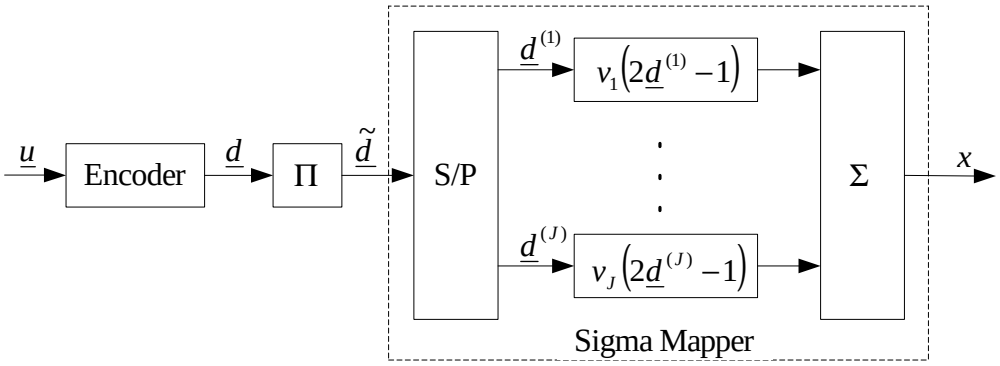


Figure 4.2 Use of sigma mapping in a coded modulation system.

parallel coded sequences. These coded sequences are then mapped to the complex symbol sequence $\underline{x}$ by the sigma mapper. The random interleaver is necessary, which makes it possible to implement a sub-optimal iterative decoding/demapping algorithm based on the well-known turbo principle [28].

4.2 An Iterative Receiver with Separate MMSE-MUD and MMSE Demodulators

The benefit of the linear property of sigma mapping and the approximate linear relationship between the transmitted symbol and the estimated one at the output of the MMSE-MUD for multiuser STBC systems is that the corresponding mapping/demapping algorithms can be implemented algorithmically instead of using table lookup as for the MAP demodulation with large signal constellations (Section 2.2.3). With the above observations an iterative MUD with separate MMSE demodulators is proposed in the following.

Specifically, with sigma mapping the transmitted symbol $c_k(l)$ (the lth symbol of the kth user) corresponding to the J coded bits $\underline{d}_k(l)$ is generated as

$$c_k(l) = \underline{V} \cdot \underline{d}_k^T(l) \tag{4.3}$$

where $\underline{d}_k(l) = [d_k(l,1), d_k(l,2), \cdots, d_k(l,J)]$ and $\underline{d}_k(l,j) \in \{\pm 1\}$.

When the MMSE-MUD and Gaussian approximation are employed, the estimated $\hat{c}_k(l)$ can be modeled as the output of an equivalent AWGN channel having $c_k(l)$ as its input symbol [9] as (see also Section 2.2.2)

$$\hat{c}_k(l) = \mu_k(l) c_k(l) + \nu_k(l) \tag{4.4}$$

Substituting (4.3) into (4.4) yields:

$$\hat{c}_k(l) = \mu_k(l) \underline{V} \cdot \underline{d}_k^T(l) + \nu_k(l) \tag{4.5}$$

By treating the real and imaginary parts in (4.5) separately, one obtains:

$$\begin{bmatrix} \text{Real}(\hat{c}_k(l)) \\ \text{Imag}(\hat{c}_k(l)) \end{bmatrix} = \begin{bmatrix} \text{Real}(H_k(l)) \\ \text{Imag}(H_k(l)) \end{bmatrix} \underline{d}_k^T(l) + \begin{bmatrix} \text{Real}(\nu_k(l)) \\ \text{Imag}(\nu_k(l)) \end{bmatrix} \tag{4.6}$$

where $H_k(l) = \mu_k(l)\underline{V}$.

The system model in (4.6) can be viewed as an equivalent multiuser system with J "virtual" users, where each "virtual user"employs antipodal signalling. It follows that the MMSE principle can also be applied for this real system (4.6). Then a low-complexity MMSE demodulator can be derived (as opposed to the complicated MAP demodulator) for each user to estimate the LLR's of the coded bits based on the estimated symbols at the output of the MUD. The difference between the MMSE demodulator and the MMSE-MUD is that the MMSE demodulator works on real quantities, whereas the MMSE-MUD works with complex symbols. The detailed derivations of the MMSE demodulator are given in Appendix C.

Replacing the MAP demodulators with the MMSE demodulators in the conventional iterative receiver, a lower-complexity iterative receiver with separate MMSE-MUD and MMSE demodulators is obtained, as shown in Fig. 4.3. For convenience, this iterative structure is referred to as MMSE-MUD/MMSE-DEM.

In the next section, the convergence property and complexity of the proposed iterative receiver are discussed.

4.3 Complexity Analysis and Convergence Property of the Proposed Iterative MMSE-MUD/MMSE-DEM Receiver

The detailed complexity analysis for the MMSE estimations of the system in (4.6) is given in Appendix C. The final average complexity of the MMSE demodulator per coded bit with sigma mapping is given by:

$$\Gamma_{\text{MMSE-DEM}} \approx n^3/3 + 3n^2 + 6n + 4 \tag{4.7}$$

With $n = 2$, one has $\Gamma_{\text{MMSE-DEM}} \approx 30$ (MULs). For the ease of comparison, recall that the complexity of the MAP demodulator per coded bit is

$$\Gamma_{\text{MAP-DEM}} = \left[\frac{(2J+3)}{J}2^J + 1\right] \text{MULs} + \frac{1}{J}2^J \text{EXPs} + 1\text{LOG} \tag{3.5}$$

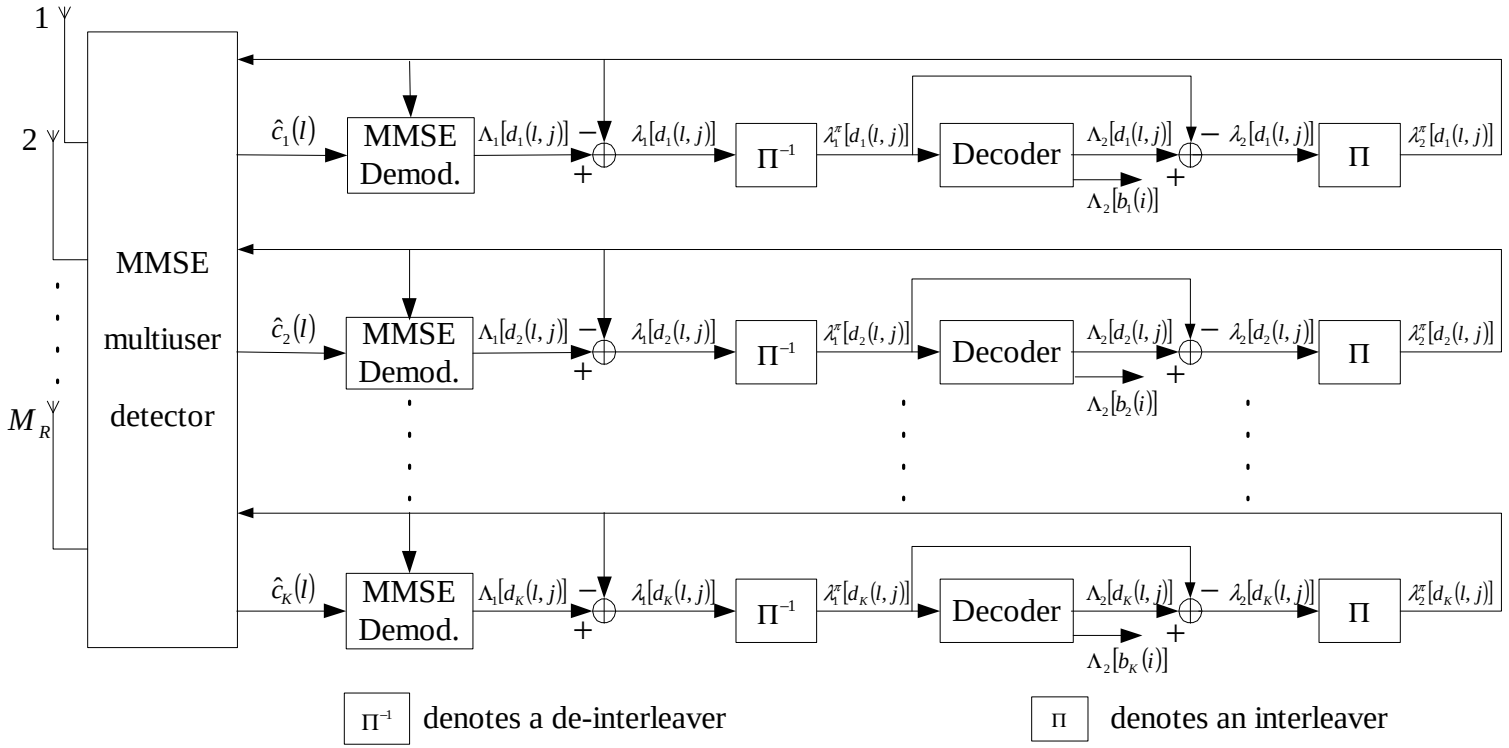


Figure 4.3 A low-complexity iterative receiver with separate MMSE demodulators.

Observe that the average complexity of the MMSE demodulator per coded bit does not depend on the constellation size when sigma mapping is employed. It always takes about 30 MULs to demodulate one coded bit, no matter how big the constellation size is. In contrast, the complexity of the MAP demodulator increases exponentially with J. For 16-QAM where $J = 4$, the complexity of the MAP demodulator is 50 (MULs) already. Here we treat EXP and/or LOG as one MUL, respectively for the simplicity of comparison. This clearly shows that when J (or equivalently M) is large the MMSE algorithm can significantly reduce the complexity of demodulator and, hence the overall system complexity, by employing sigma mapping of M-QAM.

Next, we investigate the convergence behavior of the proposed iterative receiver using the extrinsic information transfer characteristic (EXIT) charts. As done in Section 3.2.3, we combine the MUD and all the K demodulators in a single module to ease the observation of how the extrinsic information flows among the three modules (MUD, K modulators, and K decoders), shown in Fig. 4.4.

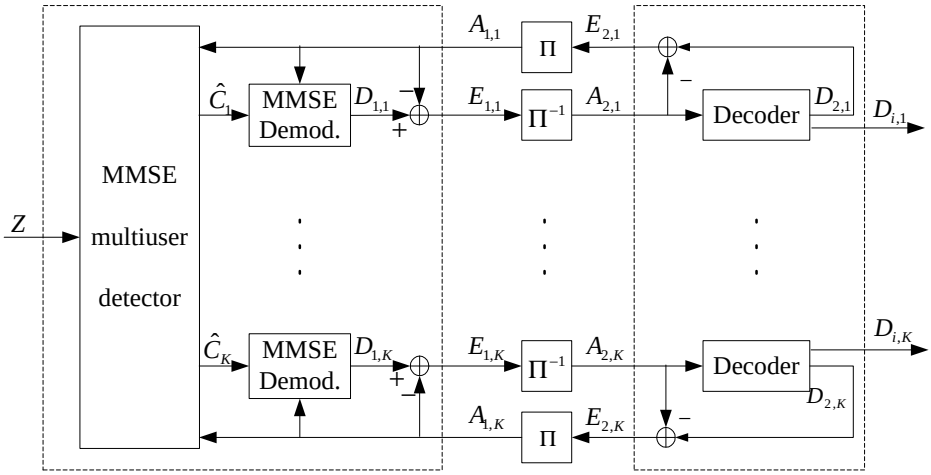


Figure 4.4 Extrinsic information flow inside the proposed receiver with separate MUD and demodulators.

Fig. 4.5 and Fig. 4.6 plot the MMSE demodulator's extrinsic information transfer characteristics for 8-QAM and 16-QAM when sigma mapping is applied, respectively. Fig. 4.7 gives the EXIT chart of the MUD with separated MMSE demodulators

for 8-QAM when it interacts with the channel decoder at 5dB. Similarly, Fig. 4.8 shows the EXIT chart of the MUD with separated MMSE demodulators for 16-QAM when interacting with the channel decoder at 8dB. Simulated trajectories of iterative decoding are also shown to demonstrate the decoding process. For comparison, the corresponding extrinsic information transfer characteristics of the conventional iterative receiver with MAP demodulator are also plotted in these figures.

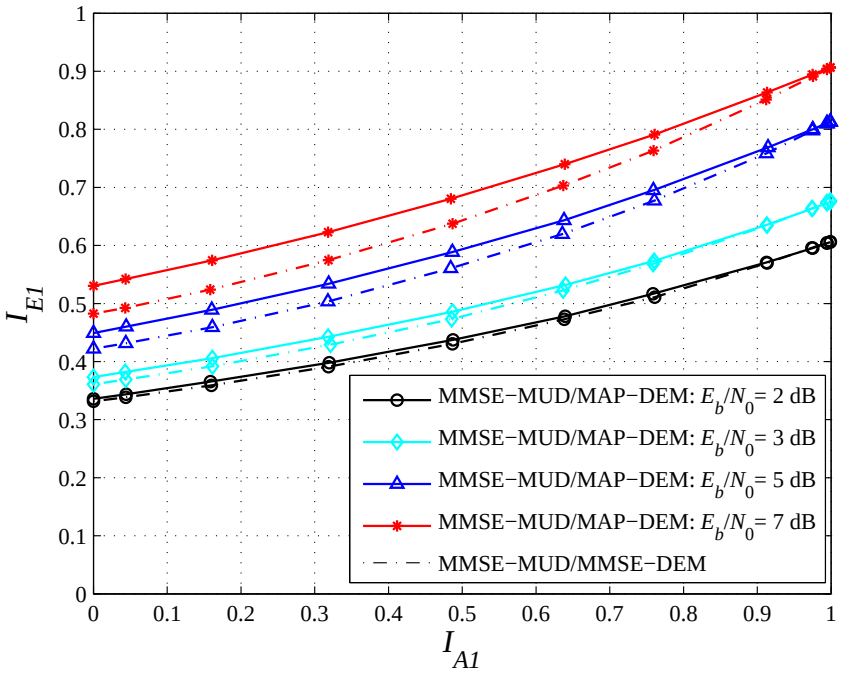


Figure 4.5 Extrinsic information transfer characteristic with separate MUD and MMSE-DEM when sigma mapping is used for 8QAM.

From Fig. 4.5 and Fig. 4.6, one can clearly see that when the *a priori* information I_{A_1} is high enough, the MUD with the low-complexity MMSE demodulator can output the same amount of the *a posteriori* information I_{E_1} as in the case of the MUD with MAP demodulators. It means that the low-complexity MMSE-MUD/MMSE-DEM receiver can approach the asymptomatic performance of MMSE-MUD/MAP-DEM very well for different levels of signal to noise ratio (E_b/N_0). This observation is promising with respect to both BER performance and the system complexity.

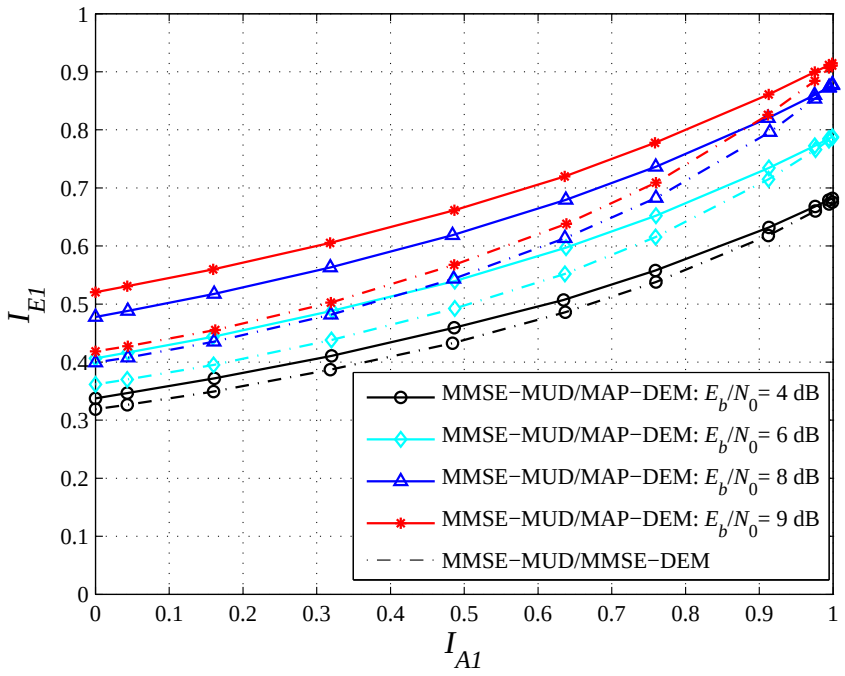


Figure 4.6 Extrinsic information transfer characteristic with separate MUD and MMSE-DEM when sigma mapping is used for 16QAM.

The disadvantage of the proposed MMSE-MUD/MMSE-DEM is that it outputs much less extrinsic information I_{E_1} when I_{A_1} is low (especially when $I_{A_1} < 0.5$). Although it does not impact the asymptomatic performance, it deteriorates the convergence property of the iterative decoding process. It can also be observed from the decoding trajectories shown in Fig. 4.7 and Fig. 4.8 that it takes more iterations for an MMSE-MUD/MMSE-DEM receiver than an MMSE-MUD/MAP-DEM receiver to achieve the same asymptomatic performance. In the next section, we will discuss how to improve the convergence property of the iterative receiver by further taking into account the advantage of sigma mapping.

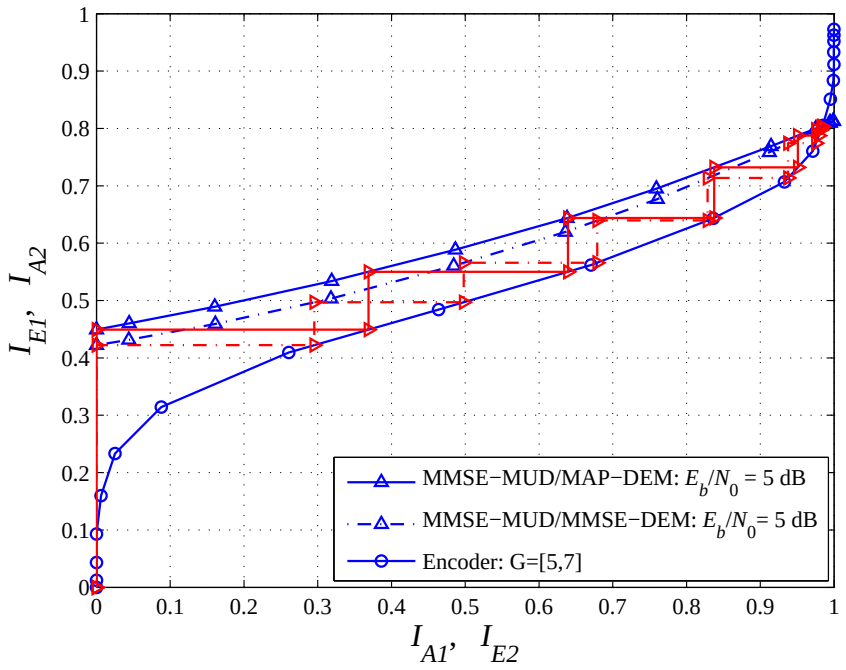


Figure 4.7 EXIT charts of iterative receivers with separate MMSE demodulators and sigma mapping of 8QAM.

4.4 An Iterative Receiver with Combined MMSE MUD and MMSE Demodulators

Let's revisit the system model in (2.15) under sigma mapping. Assume that all users employ the same basis set[1]. Substituting (4.3) into (2.15) gives:

$$\begin{aligned}\underline{\mathbf{r}} &= \mathbf{H}\cdot[\ c_1(1),\cdots,c_1(N),c_2(1),\cdots,c_2(N),\cdots,c_K(N)\ ]^T+\underline{\mathbf{n}}\\ &= \mathbf{H}\cdot[\ \underline{V}\cdot\underline{d}_1^T(1),\cdots,\underline{V}\cdot\underline{d}_1^T(N),\underline{V}\cdot\underline{d}_2^T(1),\cdots,\\ &\qquad \underline{V}\cdot\underline{d}_2^T(N),\cdots,\underline{V}\cdot\underline{d}_K^T(N)\ ]^T+\underline{\mathbf{n}}\end{aligned} \tag{4.8}$$

Reorganize (4.8), one has:

$$\begin{aligned}\underline{\mathbf{r}} = \mathbf{H}\cdot\underbrace{\mathrm{diag}(\underline{V},\underline{V},\cdots,\underline{V})}_{\text{size } NKJ\times NKJ}\cdot\ [&\underline{d}_1^T(1),\cdots,\\ &\underline{d}_1^T(N),\underline{d}_2^T(1),\cdots,\underline{d}_2^T(N),\cdots,\underline{d}_K^T(N)]+\underline{\mathbf{n}}\end{aligned} \tag{4.9}$$

[1]The framework developed here can be extended to the scenario that different users employ different basis sets.

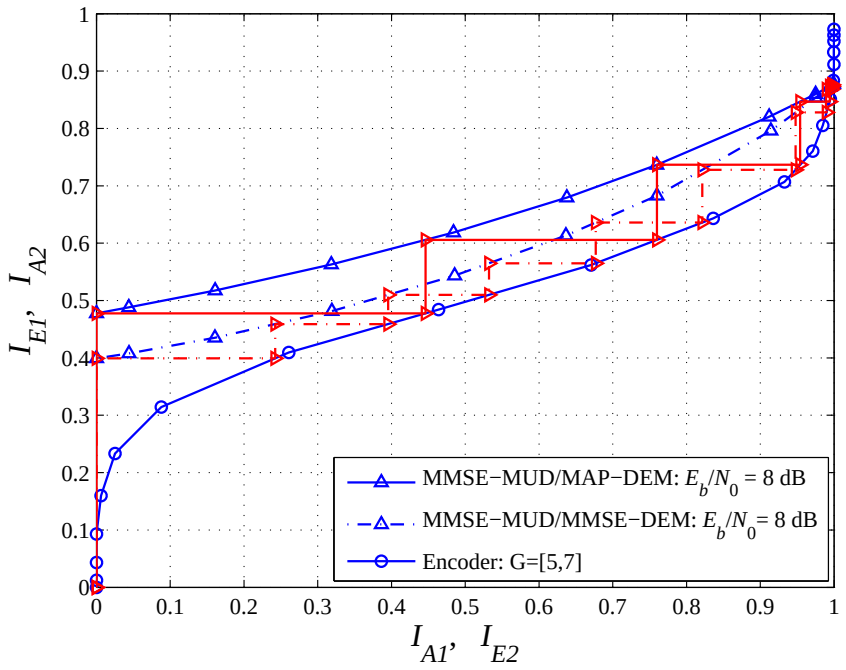


Figure 4.8 EXIT charts of iterative receivers with separate MMSE demodulators and sigma mapping of 16QAM.

Let

$$\mathbf{H}_b = \mathbf{H} \cdot \mathrm{diag}(\underline{V}, \underline{V}, \cdots, \underline{V}) \tag{4.10}$$

and

$$\underline{\mathbf{b}} \;=\; [\; \underline{d}_1(1), \cdots, \underline{d}_1(N), \underline{d}_2(1), \cdots, \underline{d}_2(N), \cdots, \underline{d}_K(N) \;] \tag{4.11}$$

Then

$$\underline{\mathbf{r}} = \mathbf{H}_b \cdot \underline{\mathbf{b}}^T + \underline{\mathbf{n}} \tag{4.12}$$

Separating the real and imaginary parts, one has

$$\begin{bmatrix} \mathrm{Real}(\underline{\mathbf{r}}) \\ \mathrm{Imag}(\underline{\mathbf{r}}) \end{bmatrix} = \begin{bmatrix} \mathrm{Real}(\underline{\mathbf{H}}_b) \\ \mathrm{Imag}(\underline{\mathbf{H}}_b) \end{bmatrix} \cdot \underline{\mathbf{b}}^T + \begin{bmatrix} \mathrm{Real}(\underline{\mathbf{n}}) \\ \mathrm{Imag}(\underline{\mathbf{n}}) \end{bmatrix} \tag{4.13}$$

Again, based on the linear system in (4.13), the soft-output MMSE algorithm can be applied to compute the extrinsic information of the coded bits *directly* from the received signals. Such a computation eliminates the need to obtain the intermediate

soft estimates of the transmitted symbols as in the conventional iterative receiver (see Fig. 4.3). In other words, the soft-output multiuser detector and the K demodulators in Fig. 4.3 can be integrated into a single module. The proposed integrated iterative receiver structure is shown in Fig. 4.9.

4.5 Complexity Analysis and Convergence Property of the Proposed Integrated Iterative Receiver

First, we investigate the convergence property of the proposed integrated receiver by examining the flow of the extrinsic information shown in Fig. 4.10. Fig. 4.11 and Fig. 4.12 give the corresponding transfer characteristic charts, which are also compared to those of the MMSE-MUD/MMSE-DEM receiver. It can be seen from the results that the integrated receiver can approach the same asymptomatic performance as that of the MMSE-MUD/MAP-DEM for all levels of E_b/N_0. Furthermore, with the same amount of *a priori* information I_{A_1}, the integrated receiver can produce more useful extrinsic information I_{E_1} when compared to the MMSE-MUD/MMSE-DEM receiver. This means that the integrated receiver can improve the convergence speed while maintaining the same asymptomatic BER performance. This improvement is due to the fact that the integrated receiver works on bit-level interference cancellation, rather than on symbol-level interference cancellation. However, compared with MMSE-MUD/MAP-DEM, the integrated receiver is still less efficient as can be seen from Fig. 4.13 and Fig. 4.14.

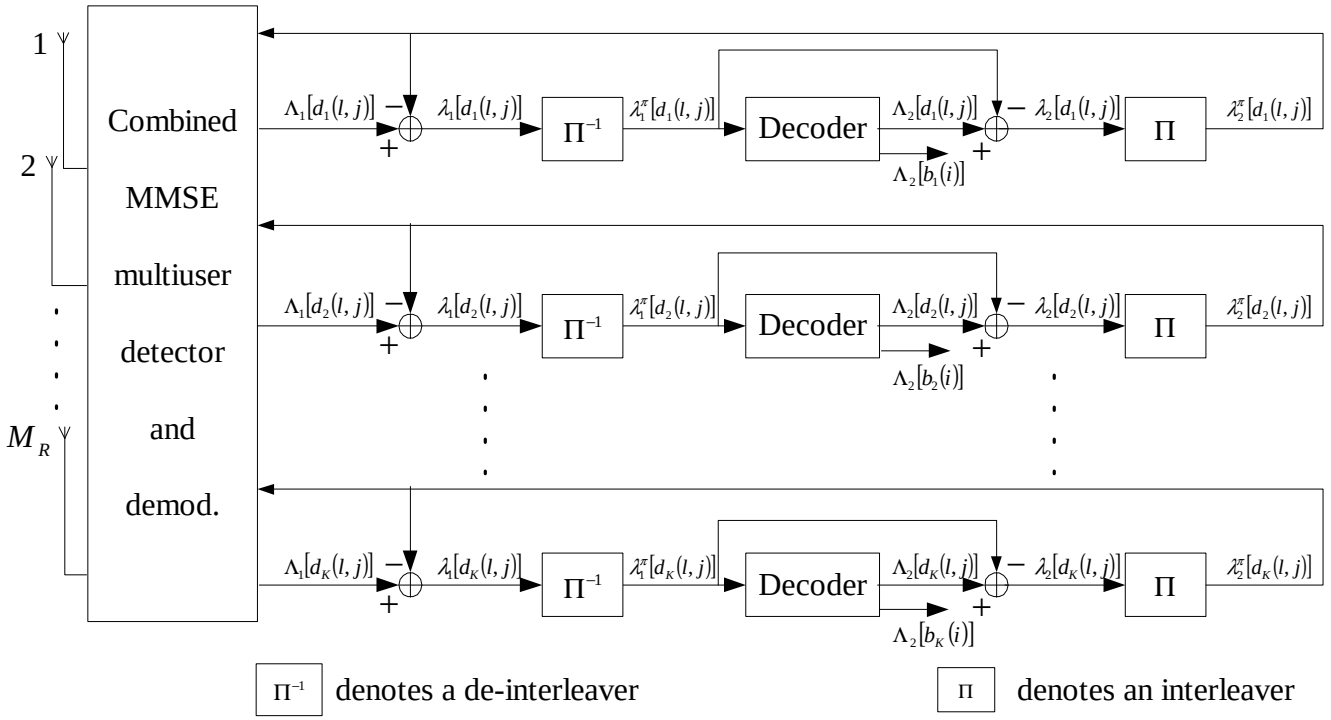


Figure 4.9 The proposed integrated iterative receives for a multiuser STBC system.

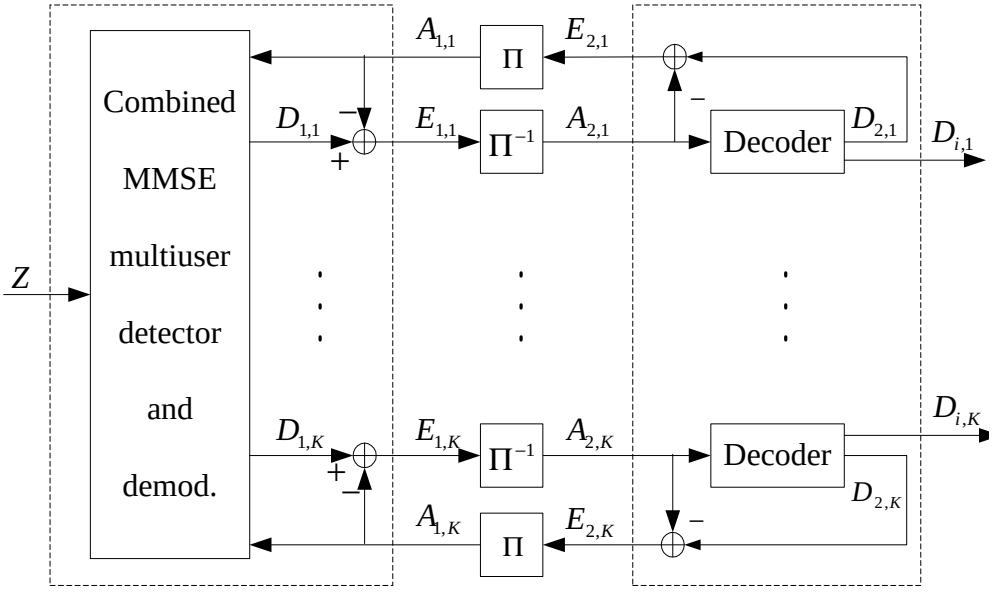


Figure 4.10 Extrinsic information flow of the proposed integrated receiver.

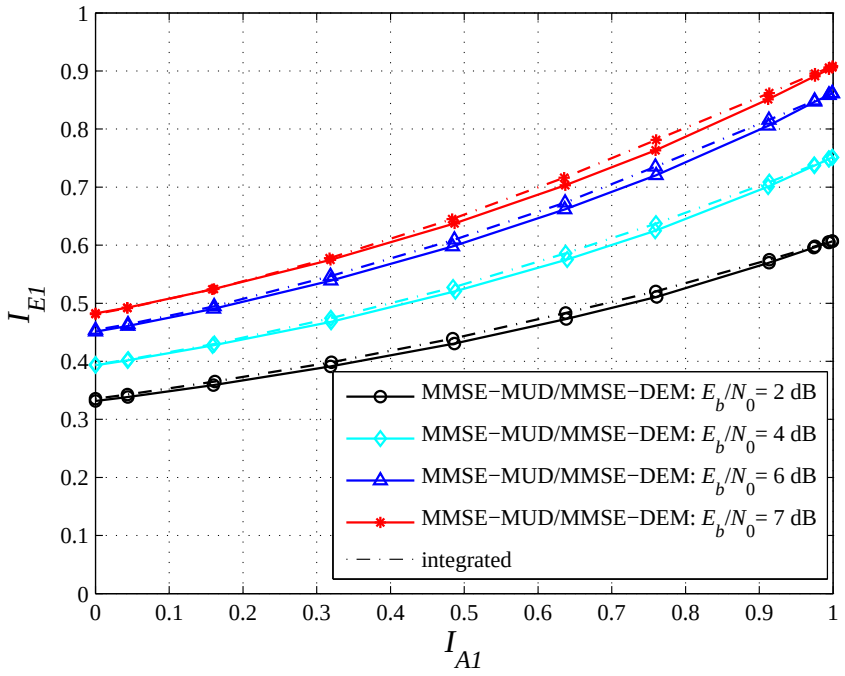


Figure 4.11 Comparison of the transfer characteristics of the separated and integrated iterative receivers: 8QAM and sigma mapping.

Next, consider the complexity of the proposed integrated receiver. According to

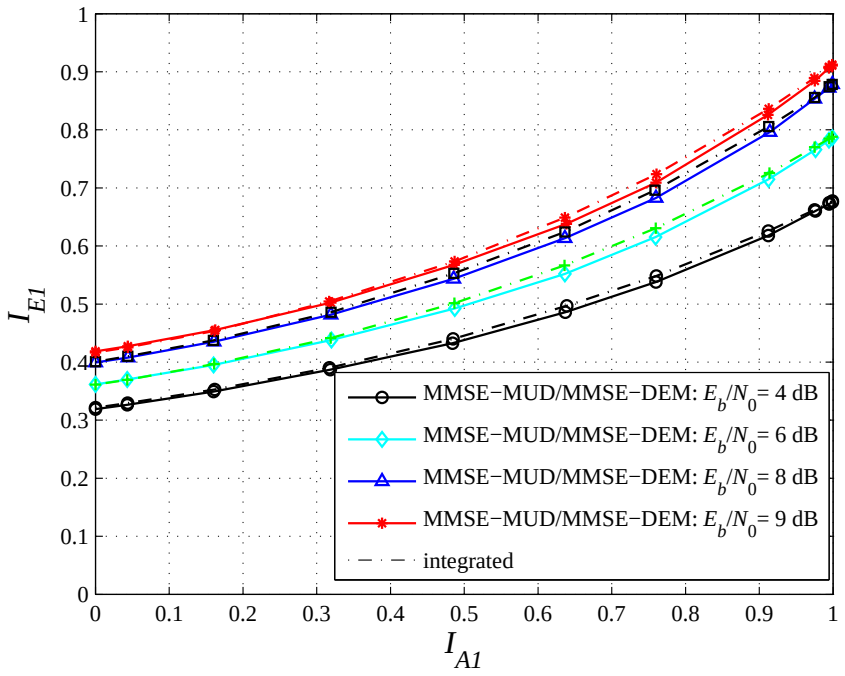


Figure 4.12 Comparison of the transfer characteristics of the separated and integrated iterative receivers: 16QAM and sigma mapping.

Appendix C, the complexity per coded bit of the integrated receiver based on (4.13) can be directly given as

$$\begin{aligned}\Gamma_{\text{integrated}} &\approx n^3/3 + 3n^2 + 6n + 4,\ n = 2M_RP \\ &= \frac{4}{3}(M_RP)^3 + 12(M_RP)^2 + M_RP + 4 \text{ (MULs)} \end{aligned} \tag{4.14}$$

The complexity of the receivers with separated MMSE demodulators and separated MAP demodulators are given as (see. Section 3.1.2 and 3.1.3)

$$\begin{aligned}\Gamma_{\text{MMSE−MUD/MMSE−DEM}} &= \frac{4}{3J}(M_RP)^3 + \frac{5}{J}(M_RP)^2 + \frac{16}{J}M_RP \\ &\quad + \frac{[3\times 2^J + 6]}{J} + 30 \end{aligned} \tag{4.15}$$

$$\begin{aligned}\Gamma_{\text{MMSE−MUD/MAP−DEM}} &= \frac{4}{3J}(M_RP)^3 + \frac{5}{J}(M_RP)^2 + \frac{16}{J}M_RP + \frac{[3\times 2^J + 6]}{J} \\ &\quad + \left[\frac{(2J+3)}{J}2^J + 1\right]\text{MULs} + \frac{1}{J}2^J\text{EXPs} + 1\text{LOG} \end{aligned} \tag{4.16}$$

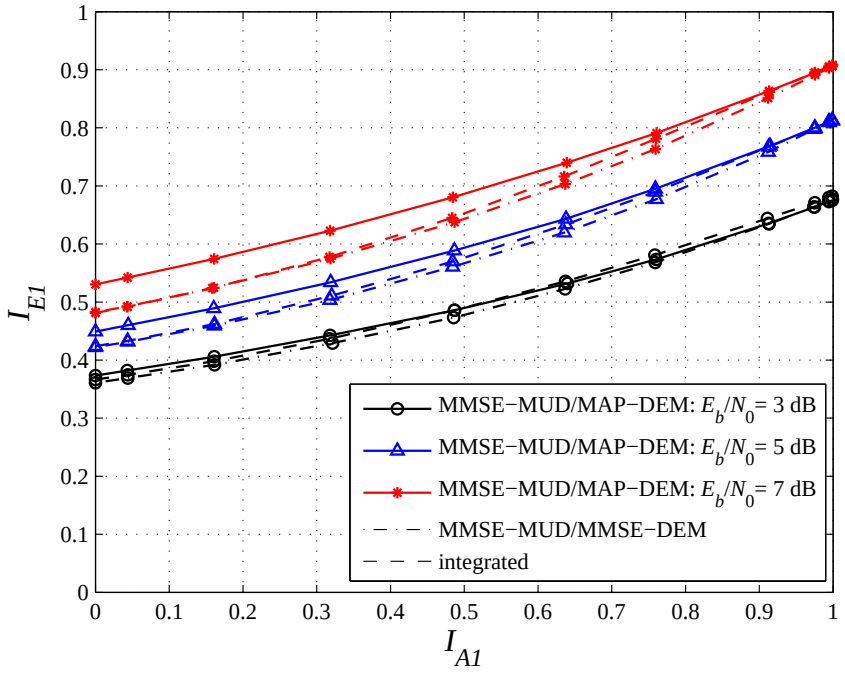


Figure 4.13 Comparison of the transfer characteristics of three MUDs for 8QAM.

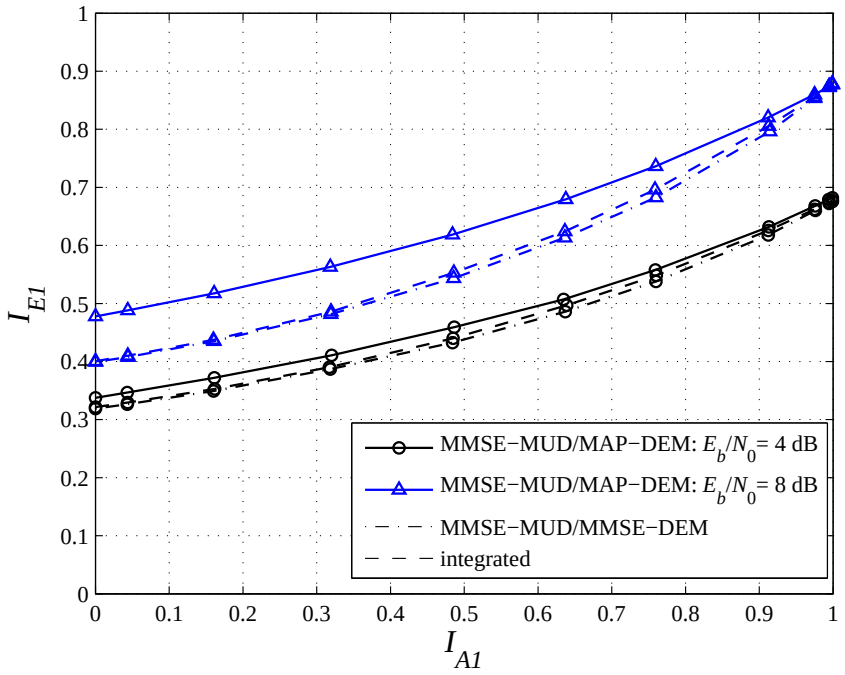


Figure 4.14 Comparison of the transfer characteristics of three MUDs for 16QAM.

Fig. 4.15 and 4.16 shows the relative complexity of the integrated and MMSE-MUD/MMSE-DEM receivers over the complexity of the MMSE-MUD/MAP-DEM (i.e., the complexity of the separated MMSE-MUD/MAP-DEM is always normalized to be 1 for any values of M_R and J). One can observe that when a small number of receive antennas and large constellation are employed (for example $M_R = 1, J = 8$ in Fig. 4.15), the integrated receiver has the lowest complexity. When more antennas are employed and the constellation size is not big ($M_R = 4, J = 6$ in Fig. 4.16), the receiver with separated MMSE MUD and MMSE demodulators has the lowest complexity.

Therefore, considering both the complexity and convergence properties, the integrated receiver is an attractive candidate to replace the MMSE-MUD/MMSE-DEM receiver when only a few receive antennas and a high order constellation are employed, which is the case of practical interest. In the next chapter, another approach is considered to improve the convergence property of the iterative receivers by investigating different iteration strategies.

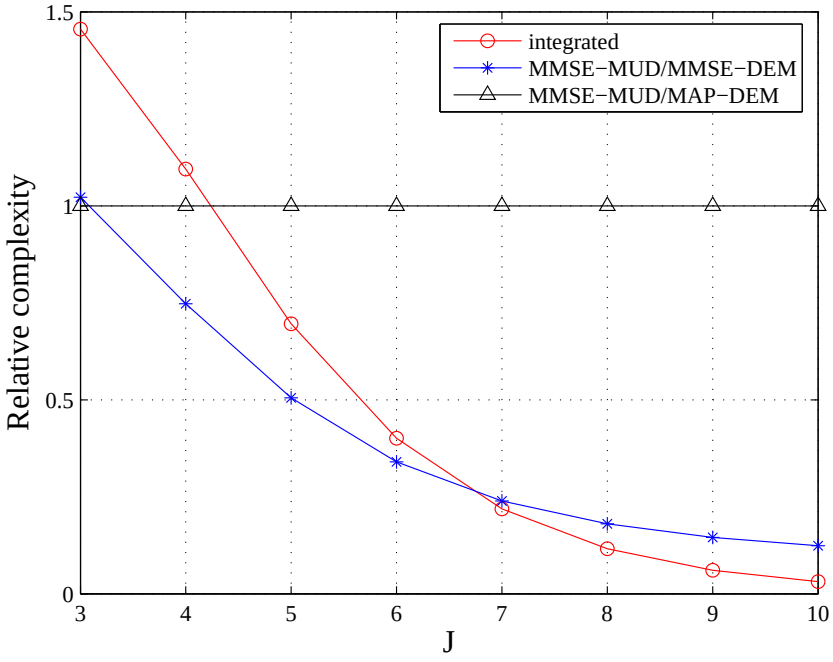


Figure 4.15 Relative complexity of three iterative receivers: $M_R = 1$

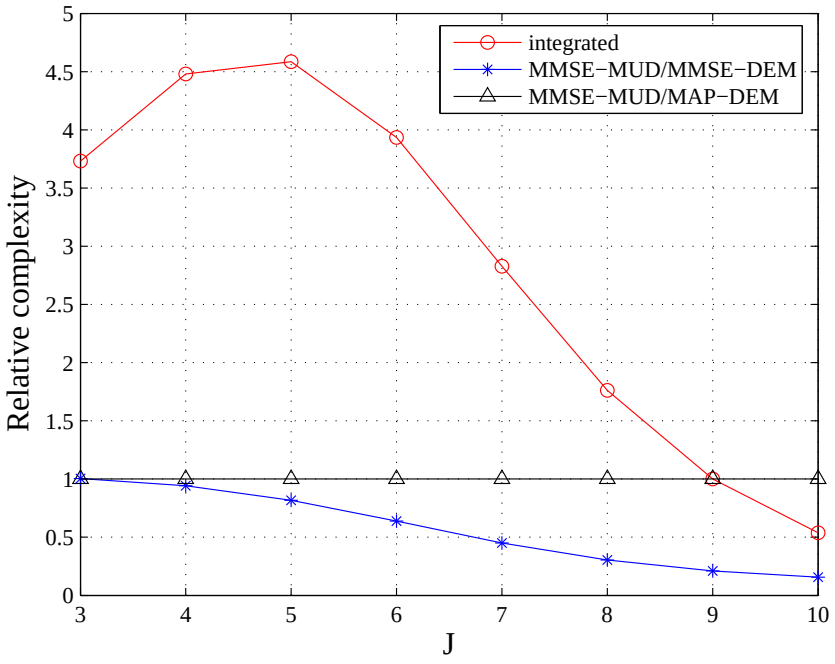


Figure 4.16 Relative complexity of three iterative receivers: $M_R = 4$

5. Two-loop Iterative Receiver for Multiuser STBC Systems

This chapter considers another way to improve the efficiency of the receivers with iterative decoding by employing different iteration strategies. A good iterative receiver depends not only on the individual modules, namely the multiuser detector, the demodulators and the decoders, but also on how effectively the information is exchanged among all the modules. In this chapter, an iterative receiver with a two-loop iteration structure for multiuser STBC systems is presented. It is shown that the proposed receiver has a much better convergence property with the iterative processing, when compared to the conventional iterative receiver, and with no increase on the system complexity.

For illustration and comparison, the signal information flow of the conventional iterative receiver is shown in Fig. 5.1 again.

Observe that the extrinsic information is exchanged in a serial manner among different modules, namely the MUD, the demodulators and the decoders. It is not clear if this serial exchange of information is optimal when multiple modules exist. Thus, the question is how to effectively operate the iterative receivers with multiple modules, i.e., how to choose the iteration strategy among the MUD, the demodulators and the decoders.

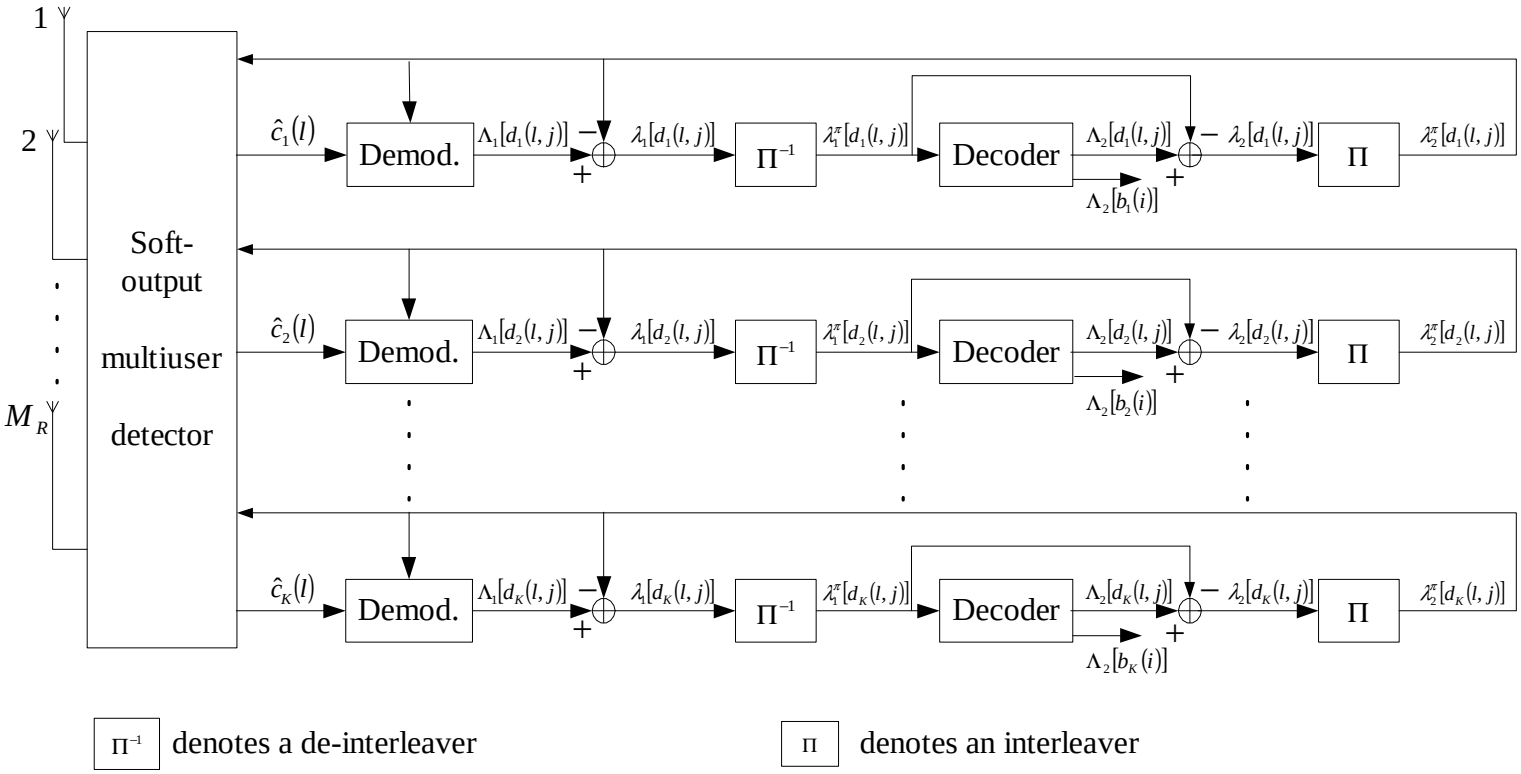


Figure 5.1 Conventional iterative receiver for a multiuser STBC system. (reproduced from Fig. 2.4

In [29], a heuristic schedule is used for operating several decoding iterations of each user with only a single cancellation iteration in uplink DS-CDMA communication systems. The cancellation module is equivalent to the MUD in a multiuser STBC system. In [30], an iterative structure with two parallel loops is proposed to accelerate the convergence of the multiple access iterative decoder.

For multiuser STBC systems, one can see from Fig. 5.1 that the MUD only outputs the estimates of the transmitted symbols $\hat{c}_k(l)$, but does not update the *a posterior* probabilities of the coded bits $\lambda[d_k(l,j)]$. Furthermore, the fact that no interleaver exists between the MUD and the demodulators makes it difficult to form an effective iteration loop for the MUD and the demodulators. However, this is not the case for the decoders and the demodulators. One can treat an M-QAM modulator as an inner encoder, whose code rate is 1. Each demodulator/decoder pair in which a random interleaver exists can form an effective inner loop to exchange the *a posteriori* information of the coded bits in addition to the outer loop which also involves the MUD. It will be demonstrated in the next section that this inner loop helps to approach the optimal decoding performance.

5.1 Iterative Receiver with Two-Loop Structure

Fig. 5.2 illustrates the iterative receiver with two-loop structure for multiuser STBC systems. The receiver consists of a soft-output multiuser detector (MUD), followed by K parallel soft-output demodulators and MAP convolutional decoders. The demodulator and the channel decoder in one branch are separated by interleaver and deinterleaver. Observe that there is an outer loop and an inner loop to perform the exchange of the extrinsic information among the soft-output MUD, the soft-output M-ary demodulators, and the SISO channel decoders.

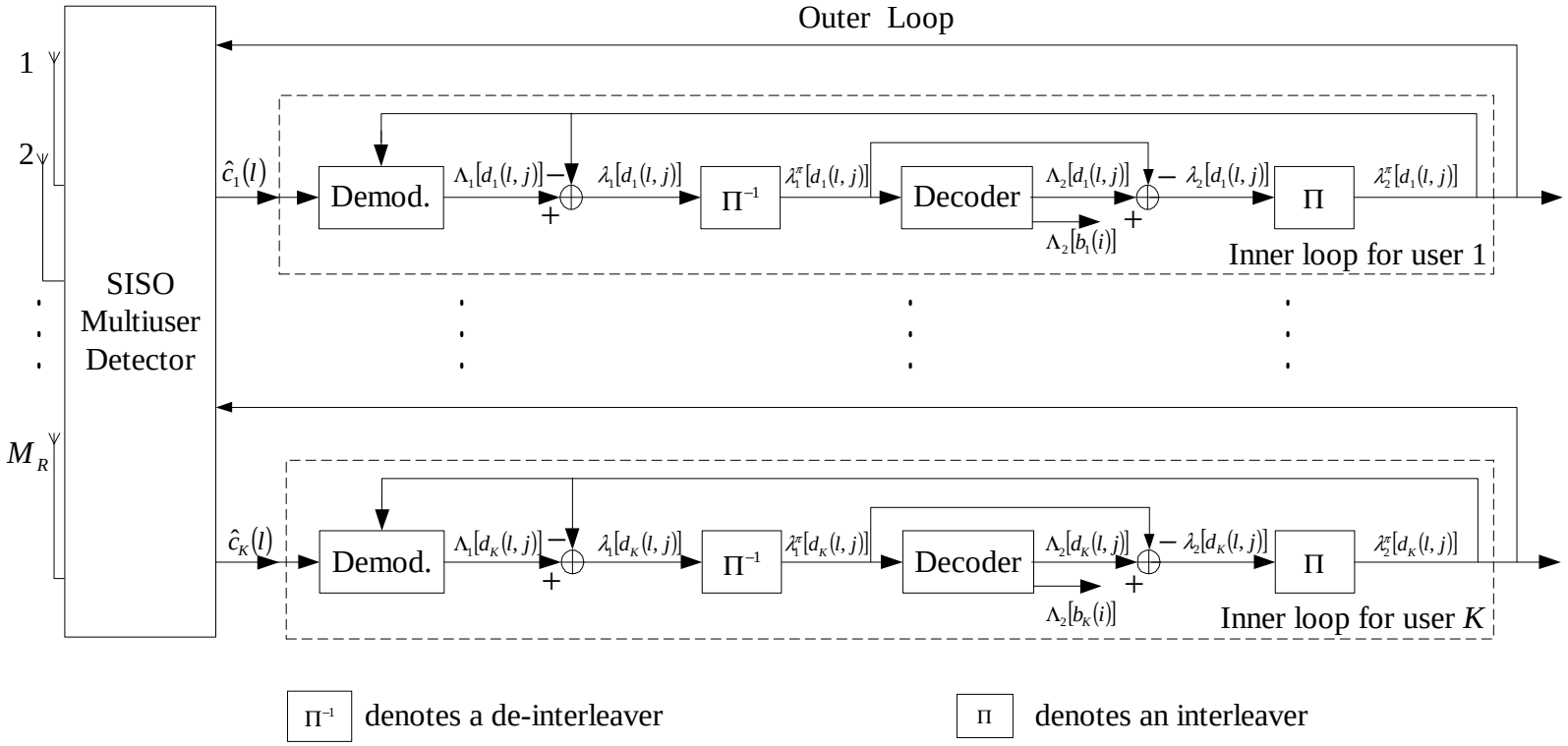


Figure 5.2 An iterative receiver with two-loop structure for a multiuser STBC system.

For the outer loop, the soft-output MUD takes as its input the received signal from M_R receive antennas and the interleaved extrinsic LLRs of the coded bits $\{\lambda_2^{\pi}[d_k(l,j)]\}, k = 1, \cdots, K$ from the last iteration of each user's inner loop, which are provided from K users' SISO decoders. The MUD provides as its output the soft estimation of the channel symbols $\hat{c}_k(l)$ for all users.

For the inner loop of the kth user, the soft-output M-ary demodulator takes as its input the soft estimate of channel symbol $\hat{c}_k(l)$ delivered by the soft-output MUD and the *a priori* extrinsic log-likelihood ratios (LLRs) $\{\lambda_2^{\pi}[d_k(l,j)]\}$ of the coded bits from the corresponding channel decoder. It produces as its output the *a posteriori* extrinsic LLRs $\{\lambda_1[d_k(l,j)]\}$ of the coded bits. Based on the interleaved extrinsic LLRs $\{\lambda_1^{\pi}[d_k(l,j)]\}$ and the code constraint of the convolutional code, the channel decoder outputs the updated extrinsic LLRs $\{\lambda_2[d_k(l,j)]\}$. The channel decoder also provides the LLRs $\{\Lambda_2[b_k(i)]\}$ of the information bits so that the information bits can be decoded at each iteration if needed. At the very first iteration, no *a priori* information is available, thus $\lambda_2^{\pi}[d_k(l,j)] = 0$.

It can be seen that the conventional iterative receiver is just a special case of the proposed receiver when only one inner loop iteration is implemented for every outer loop iteration. The special case is by no means optimal. In order to achieve a better convergence property of the iterative receiver, the inner loop for every user should be executed more than once for each outer loop iteration. Note that executing the inner loops more than one time only slightly increases the overall computational complexity when the same number of outer loop iterations are executed. This is because the inner loops do not involve the soft-output MUD, which is identified to be the most complicated block in the receiver, especially when the number of users and/or the number of transmit/receiver antennas are large. In addition, the inner loops can be executed simultaneously (i.e., in parallel) for each outer loop, and hence reducing the decoding delay.

5.2 Complexity Analysis and Convergence Property of the Two-Loop Iterative Receiver

The complexity of the MMSE multiuser detector, the MAP channel decoder, the MAP demodulator and the MMSE demodulator are approximately given as

$$\Gamma_{\text{MMSE-MUD}} \approx \frac{4}{3J}(M_R P)^3 + \frac{5}{J}(M_R P)^2 + \frac{16}{J} M_R P \tag{5.1}$$

$$\Gamma_{\text{MAP-DEC}} \approx 2^{n_0} + \frac{(2n_0 + 2 + 2^{1-k_0})}{n_o} 2^{k_o \nu} (\text{MULs}) \tag{5.2}$$

$$\Gamma_{\text{MAP-DEM}} \approx \left[\frac{(2J+3)}{J} 2^J + 1\right] \text{MULs} + \frac{1}{J} 2^J \text{EXPs} \tag{5.3}$$

$$\Gamma_{\text{MMSE-DEM}} \approx 30(\text{MULs}) \tag{5.4}$$

Consider a simple but typical 2×2 Alamouti STBC scheme ($N = 2$ and $P = 2$) [6]. Each space-time block codeword contains two complex symbols, which are transmitted over the channel in two consecutive time slots. Assume that there are four users ($K = 4$) in the system and four antennas ($M_R = 4$) are deployed at the receiver. For each user, 16-QAM constellations and a 4-state, rate-1/2 convolutional code with generator polynomial $(5, 7)$ are employed. For this configuration ($M_R = 4, P = 2, J = 4, k_0 = 1, n_0 = 2, \nu = 2$), the complexity $\Gamma_{\text{MMSE-MUD}}$ is more than 283 (MULs). However for inner-loop total complexity $\Gamma_{\text{MAP-DEC}} + \Gamma_{\text{MAP-DEM}}$ is only about 68(MULs). When sigma mapping is employed, the inner-loop complexity $\Gamma_{\text{MAP-DEC}} + \Gamma_{\text{MAP-DEM}}$ is even less, only about 48 (MULs) for any constellation size. So running an additional iteration for the inner-loop does not significantly impact the computation burden of the whole receiver because the overall computational complexity is dominated by the soft-output MUD which involves only the outer loop iterations.

Next, we investigate the convergence property of the proposed two-loop iterative receiver with extrinsic information transfer characteristics. Fig. 5.3 shows the extrinsic information flow of the proposed receiver. Observe again that there is an inner loop implemented in the MUD/DEMO block, which is not the case with the conventional receiver shown in Fig. 3.3.

Fig. 5.4 shows the signal constellations for 16-QAM with SSP mapping and sigma

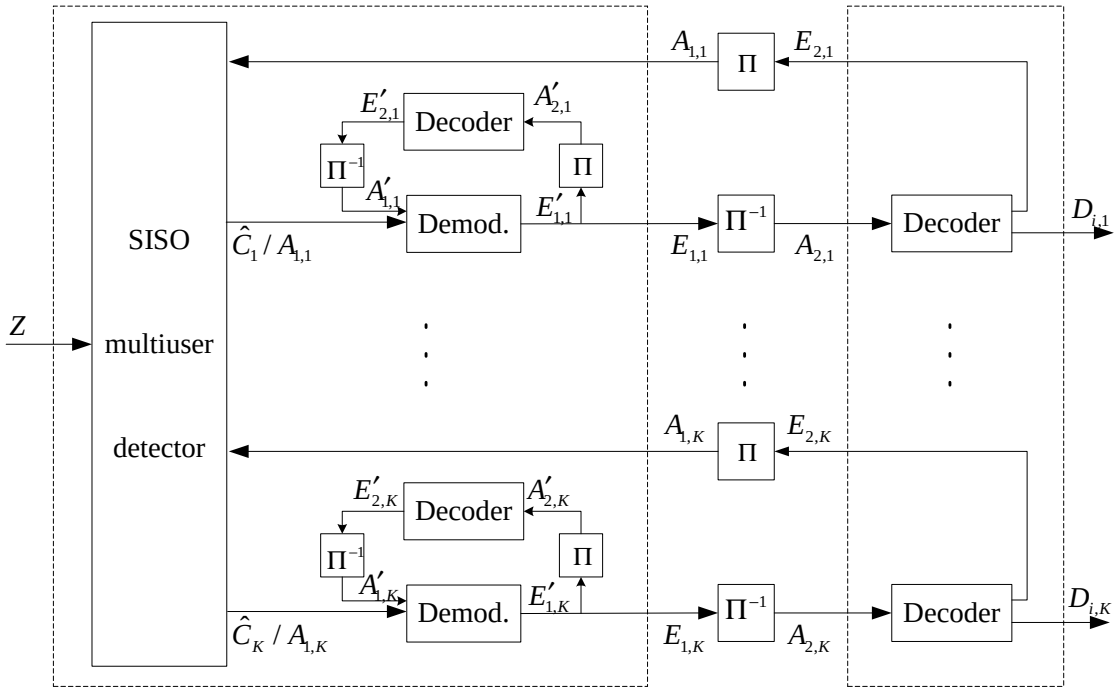
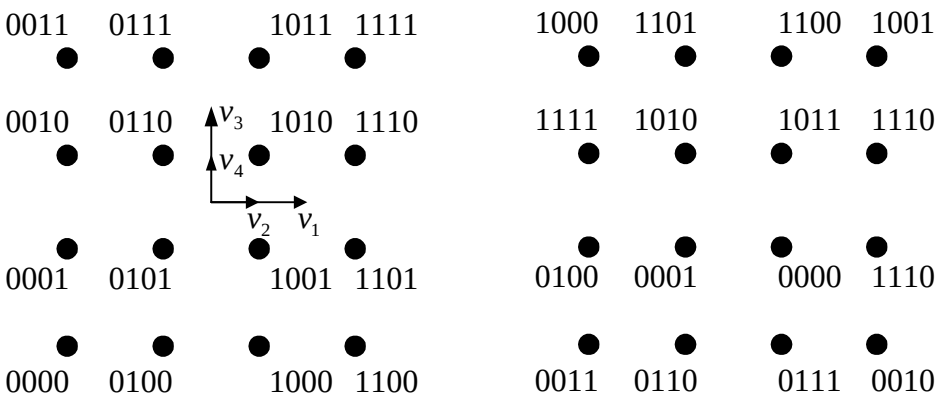


Figure 5.3 Extrinsic information flow of the proposed two-loop iterative receiver.

mapping. Figs. 5.5, 5.6, and 5.7 plot the extrinsic information transfer characteristics of MUD/DEMOs for the following cases: SSP mapping with MAP demodulators, sigma mapping with MAP demodulators and sigma mapping with MMSE demodulators, respectively. All the transfer characteristics are also compared to those of the conventional receivers in the same figure.

From the results, one can clearly see that the transfer characteristic curve of the MUD/DEMOs are lifted up at all E_b/N_0 levels when comparing the proposed receiver with the conventional receiver. This means that with the same amount of the *a priori* information I_{A_1}, running an addition inner-loop iteration as proposed can output much more useful extrinsic information I_{E_1} than that of the conventional receiver. This observation implies a better iterative convergence behavior, meaning that the asymptomatic performance can be approached with a smaller number of outer-loop iterations for the proposed iterative receiver when compared with the conventional receiver. This is confirmed by tracing the trajectories shown in Figs. 5.8, 5.9, and 5.10 for $E_b/N_0 = 8$dB. Approaching the same error performance with a smaller number of



(a) Sigma Mapping for 16-QAM (b) SSP Mapping for 16-QAM

Figure 5.4 Sigma mapping with $v_1 = \sqrt{1.6}\exp(j0)$, $v_2 = \sqrt{0.4}\exp(j0)$, $v_3 = \sqrt{1.6}\exp(j\frac{\pi}{2})$ and $v_4 = \sqrt{0.4}\exp(j\frac{\pi}{2})$ and SSP mapping for 16-QAM.

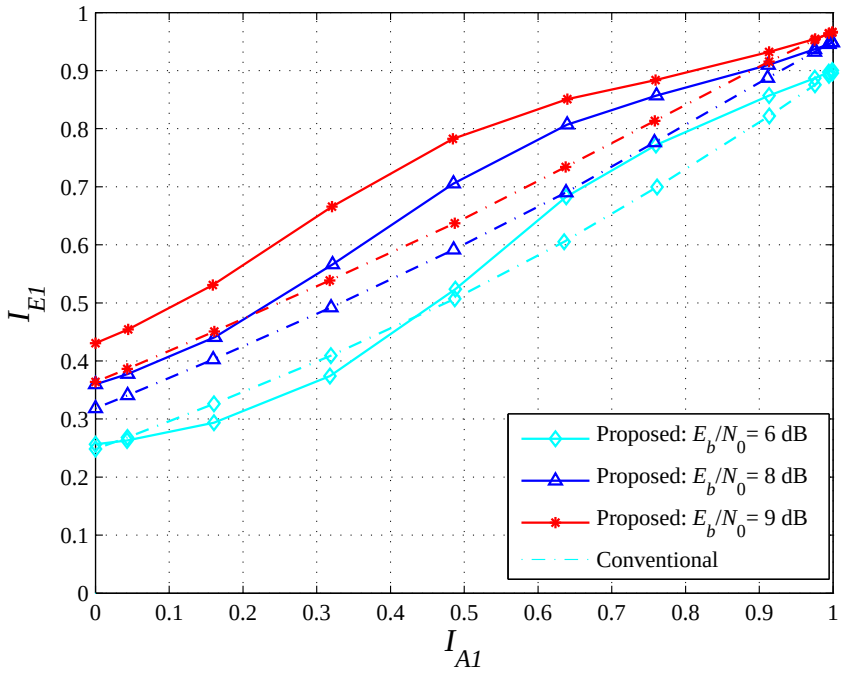


Figure 5.5 Comparison of transfer characteristics for 16-QAM with SSP mapping and MAP demodulation.

outer-loop iterations means that the receiver's convergence property is improved.

Having shown that running an additional inner-loop iteration is advantageous,

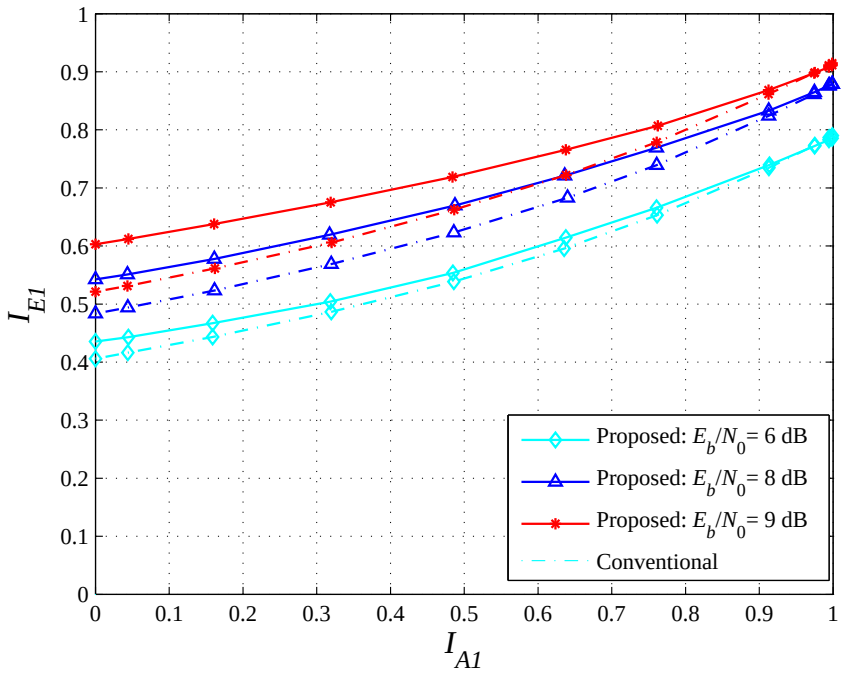


Figure 5.6 Comparison of transfer characteristics for 16-QAM with sigma mapping and MAP demodulation.

the next question is how many inner-loop iterations should be executed for one outer loop iteration when taking both convergence behavior and receiver complexity into account? This can be answered by investigating how the extrinsic information transfer characteristic evolves with each inner-loop iteration.

Systems with 16-QAM and different mapping and modulation schemes are chosen again as examples to show how to determine the most suitable number of inner-loop iterations. According to the extrinsic information flow of the proposed two-loop iterative receiver shown in Fig. 5.3, we can investigate how the output I_{E_1} can be improved with the same amount of I_{A_1} by introducing the inner-loop iterations. Figs. 5.11, 5.12, and 5.13 demonstrate the results for each scheme when $E_b/N_0 = 8$dB.

From these figures, one can observe that the output extrinsic information I_{E_1} is increased significantly with inner-loop iterations when the same amount of input extrinsic information I_{A_1} is provided, especially when I_{A_1} is not so high ($I_{A_1} < 0.85$).

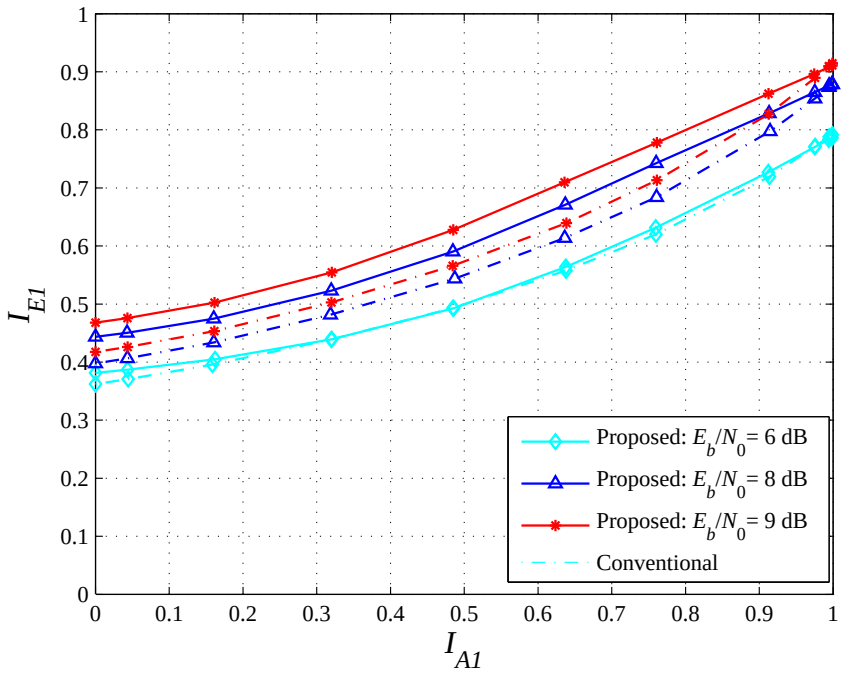


Figure 5.7 Comparison of transfer characteristics for 16-QAM with sigma mapping and MMSE demodulation.

More reliable extrinsic information I_{A_1} allows the systems to approach the asymptomatic performance in a faster way. One can also see that the iteration gain for I_{E_1} decreases as the inner-loop iteration number increases. For the first few inner-loop iterations, the convergence property is improved to a great extent while the complexity is not increased significantly. This suggests the use of two or three inner-loop iterations runs for every outer-loop iteration in the systems using 16-QAM and sigma mapping. Following the same procedure, it should not be difficult to determine the best scheme for other systems using different configurations.

5.3 Effect of Using Phase Offsets for Different Users on System Performance

The effect of using phase offsets for different users on multiuser STBC system performance is analyzed by investigating the extrinsic transfer characteristic. Here all four users employ 16-QAM with sigma mapping. The first user's signal constellation

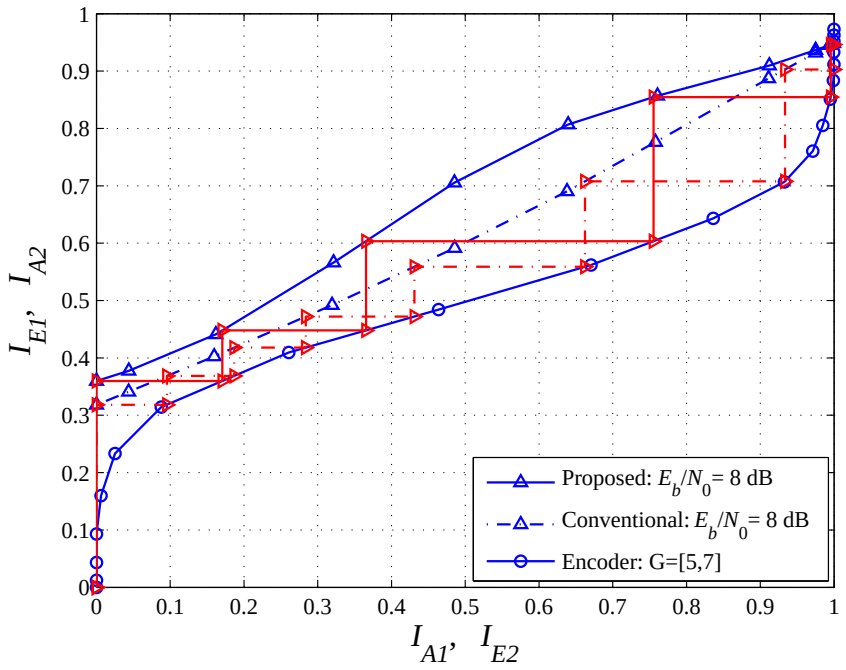


Figure 5.8 Iterative decoding trajectories for 16-QAM with SSP mapping and MAP demodulation.

is the same as the one shown in Fig. 5.4-(a). Other users' constellations are obtained by clockwise rotating the first user's constellation with fixed amount of phase offsets. Without loss of generality, phase offsets are chosen as $0, \pi/8, \pi/4$ and $\pi/2$. Fig. 5.14 demonstrates the corresponding extrinsic information transfer characteristics. The proposed two-loop iterative receiver is adopted for iterative decoding, where two inner-loop iterations are executed for one outer-loop iteration.

One can observe that the transfer characteristics for different phase offsets are almost the same. This means that for multiuser coded STBC systems, BER performances are not affected by the phase offsets among users' signal constellations. Note that a similar phenomenon was observed for coded multiple access systems with QPSK modulation and iterative decoding under an AWGN channel in [31].

For multiuser STBC systems operating over a Rayleigh fading channel, this observation can be explained as follows: the original constellation at transmitters are

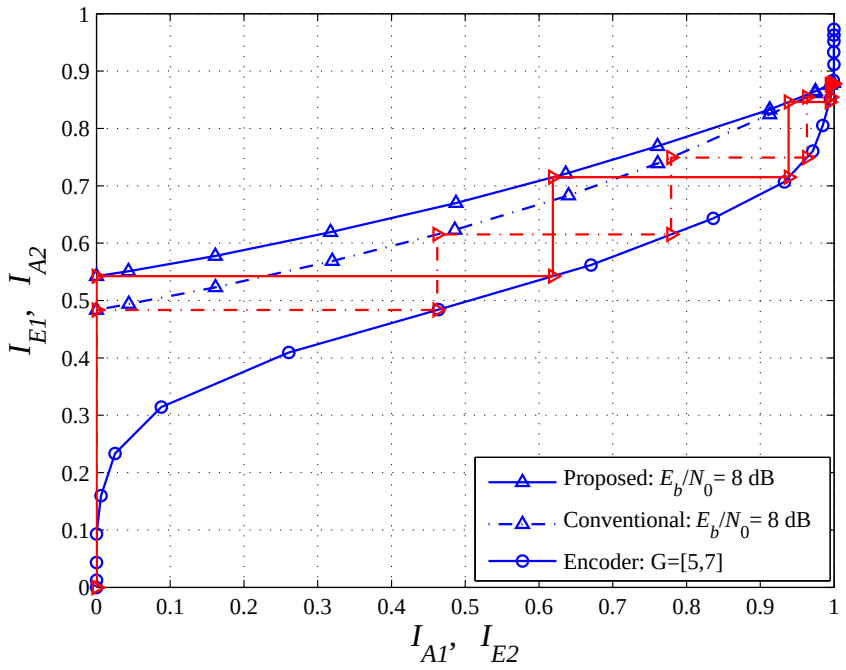


Figure 5.9 Iterative decoding trajectories for 16-QAM with sigma mapping and MAP demodulation.

randomly shaped by the channel fading gain, which distributes as the circular additive white Gaussian noise, so at the receiver the received signal for all users are distributed across the whole two-dimensional complex plain without difference and do not depend on the initial phase offset at the transmitter. Therefore, all users can employ the same constellation set for simplicity without loss of BER performance.

5.4 Simulation Results and Discussions

This section shows how BER performances can be improved by the proposed two-loop iterative receiver. Without loss of generality and for an affordable simulation time, a simple but typical 2×2 Alamouti STBC scheme ($N = 2$ and $P = 2$) [6] is applied for a four-user STBC system as in previous sections. Assume that four antennas ($M_R = 4$) are deployed at the receiver. For each user, 16-QAM constellation and a 4-state, rate-1/2 convolutional code with generator polynomial $(5, 7)$ are employed. Each user has a unique random bit-interleaver of length $L = 12,000$ coded

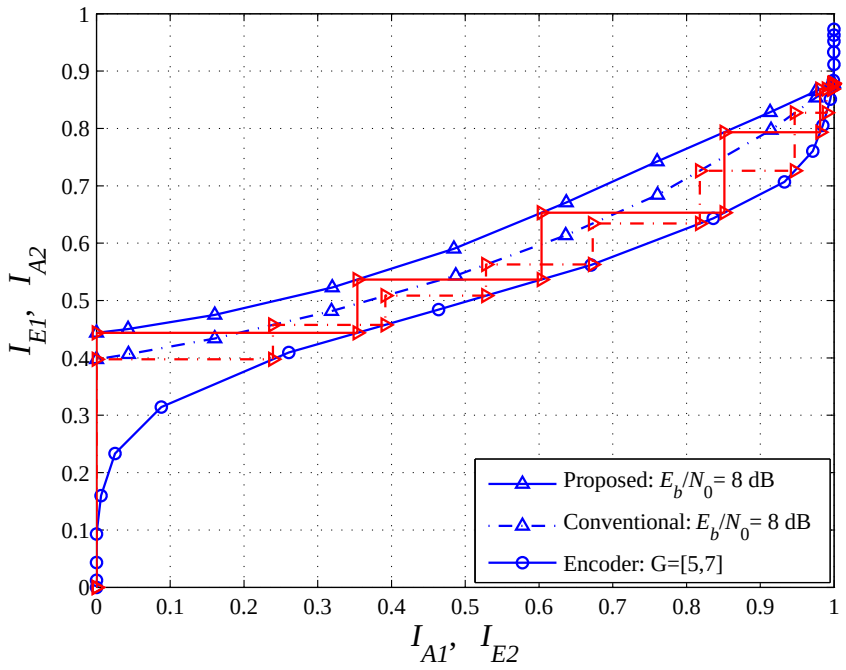


Figure 5.10 Iterative decoding trajectories for 16-QAM with sigma mapping and MMSE demodulation.

bits. This long interleaver is to make sure the independent assumption of the extrinsic information of the coded bits during iterative decoding.

Regarding the definition of the signal-to-noise ratio when multiple antennas exist, E_b/N_0 here is defined as the ratio of the total signal energy collected from all receive antennas per information bit to the noise power spectral density at the receiver [32].

First, we show the performance improvement achieved by the proposed receiver by running two iterations of the inner loop for every one iteration of the outer loop. The BER performances are illustrated in Fig. 5.15 for 16-QAM with sigma mapping and in Fig. 5.16 for 16-QAM with SSP mapping, respectively. Both cases employ MAP demodulators. Here, up to 6 iterations are performed for the conventional receiver, while a maximum number of 3 iterations of the outer loop are run for the proposed receiver.

It can be clearly observed from Fig. 5.15 and Fig. 5.16 that the proposed re-

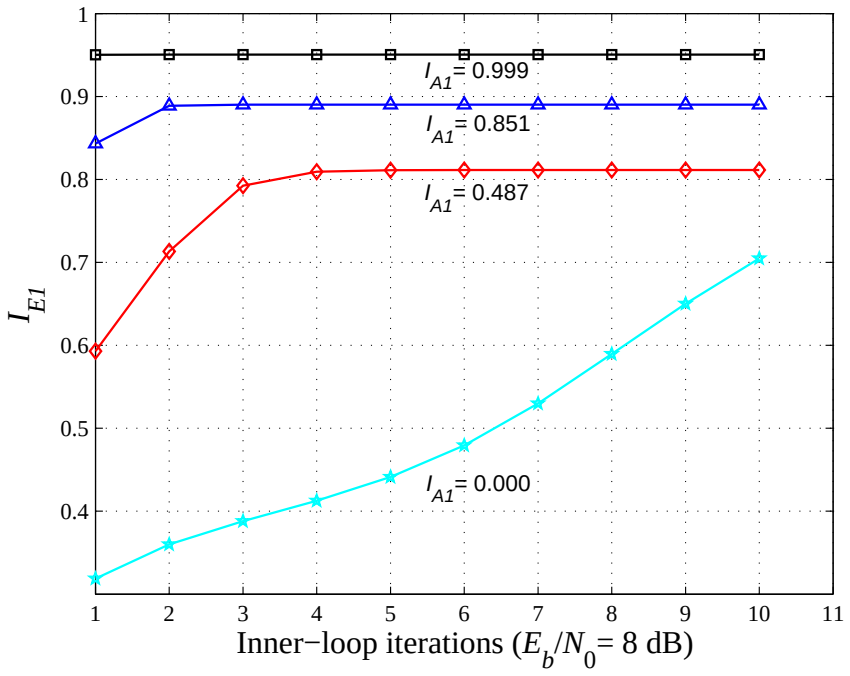


Figure 5.11 Effects of inner-loop iterations on the extrinsic information transfer characteristic: SSP mapping with MAP demodulation.

ceiver can outperform the conventional receiver at any number of outer iterations for both mapping schemes. Running two inner-loop iterations therefore significantly improve the convergence of the iterative decoding in approaching the asymptomatic performance.

Next, the sigma mapping of 16-QAM is considered. The following iterative receivers are investigated and compared: (i) The conventional iterative receiver but with MMSE demodulators instead of the complicated MAP demodulators as in [9]; (ii) The proposed iterative receiver where two inner-loop iterations are executed for every one outer-loop iteration; and (iii) The integrated iterative receiver discussed in Chapter 4 where multiuser detector and user demodulators are combined into a single module.

Fig. 5.17 plots the BER curves for the first two iterative receivers. It can be seen again here that the proposed iterative receiver outperforms the conventional one at

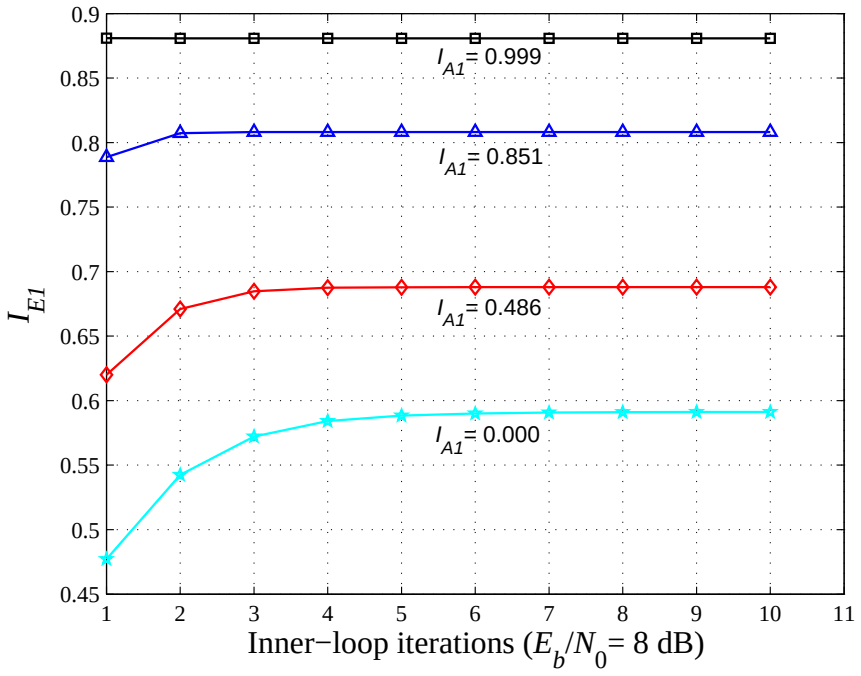


Figure 5.12 Effects of inner-loop iterations on the extrinsic information transfer characteristic: sigma mapping with MAP demodulation.

any number of outer loop iterations. The BER performance after 4 iterations of the proposed receiver already approaches the BER performance of the conventional one after 7 iterations. These results show that the convergence of the iterative processing is significantly improved with the proposed receiver.

Next, Fig. 5.18 compares the performances of the proposed receiver with that of the integrated one. Also shown in this figure is the BER performance after 8 iterations of the conventional receiver[1] to serve as the lower bound. Clearly, with the same number of outer loop iterations, the iterative receiver proposed in this chapter also outperforms the integrated iterative receiver.

The outstanding performance improvement of the proposed receiver comes at the expense of a reasonable increase in computational complexity of one outer-loop iteration by running an additional inner-loop iteration. However the reduced iterative

[1]This corresponds to the highest computational complexity.

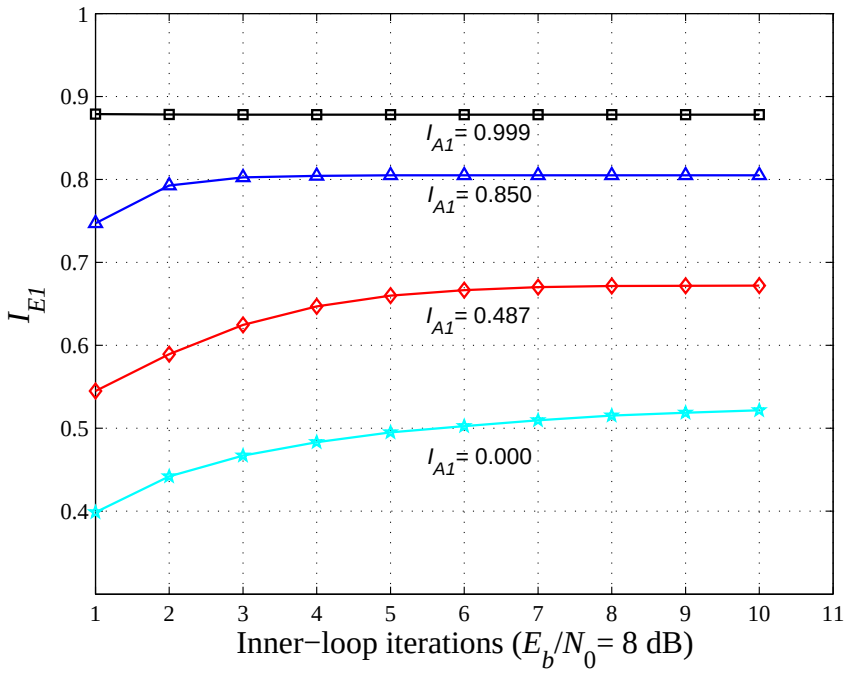


Figure 5.13 Effects of inner-loop iterations on the extrinsic information transfer characteristic: sigma mapping with MMSE demodulation.

times from outer-loop greatly decrease the whole system complexity in turn. It is very attractive when taking into account the error performance, receiver complexity and decoding delay.

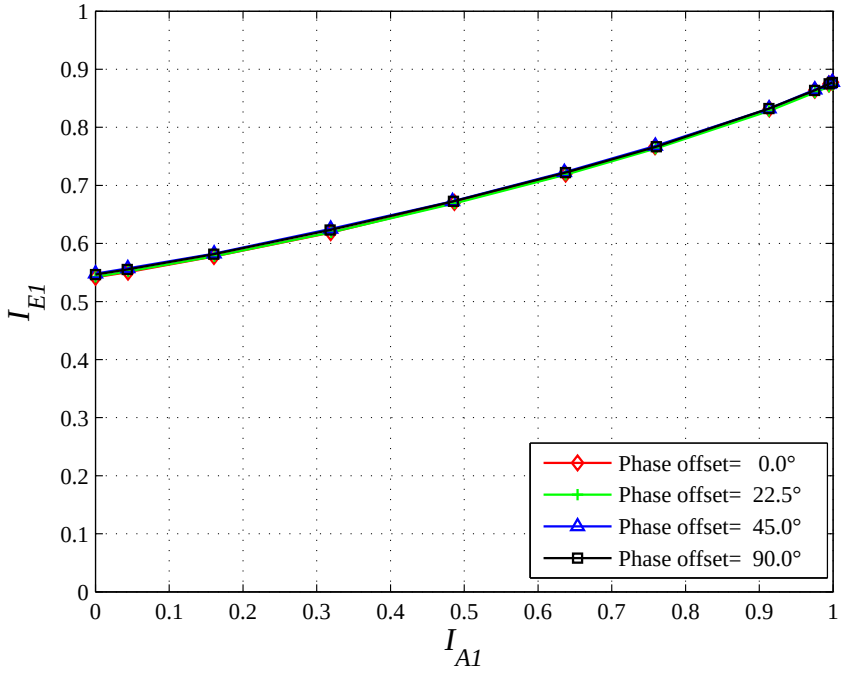


Figure 5.14 Effects of using different phase offsets for four users' constellations: 16-QAM/sigma mapping with MAP demodulation.

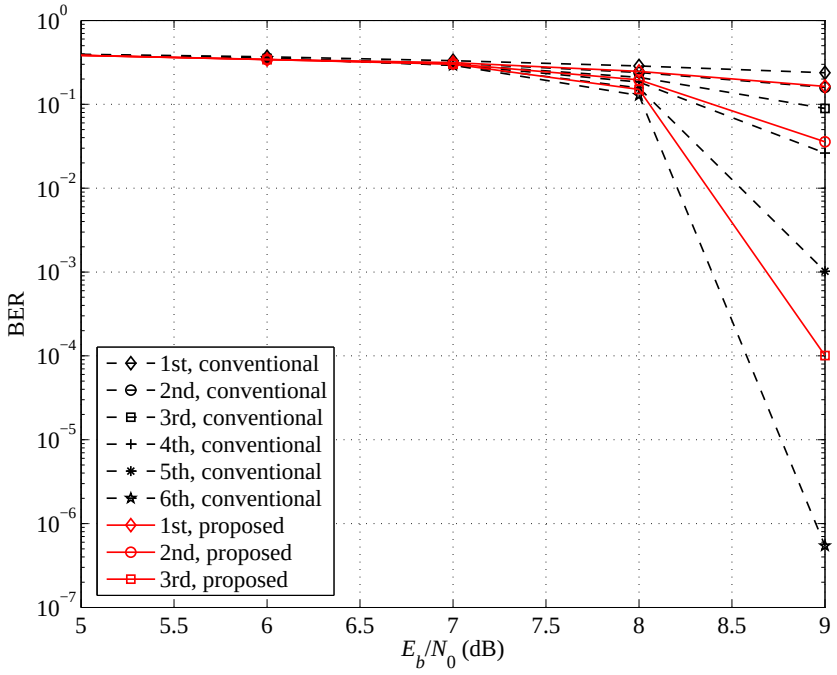


Figure 5.15 BER performance with SSP mapping and MAP demodulator.

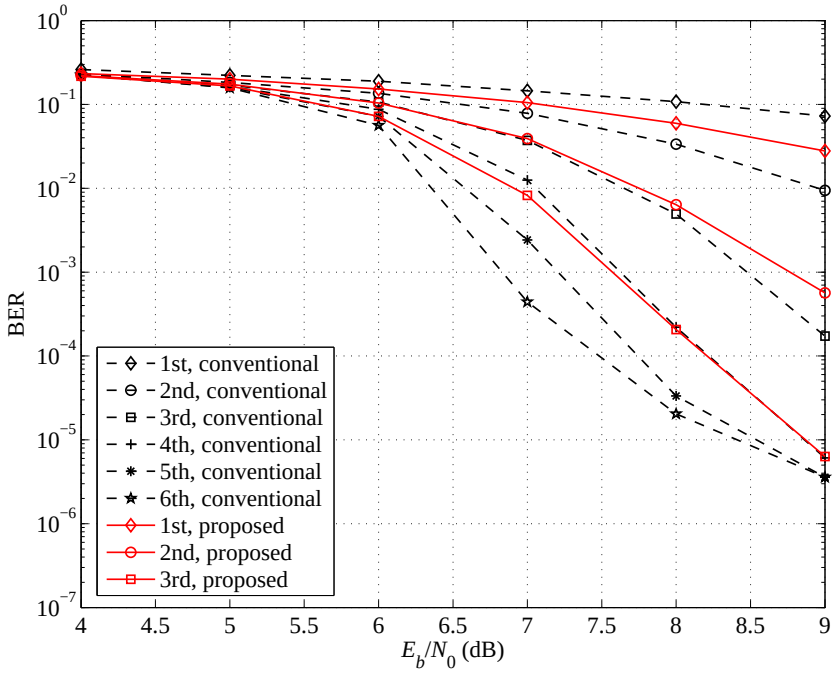


Figure 5.16 BER performance with sigma mapping and MAP demodulator.

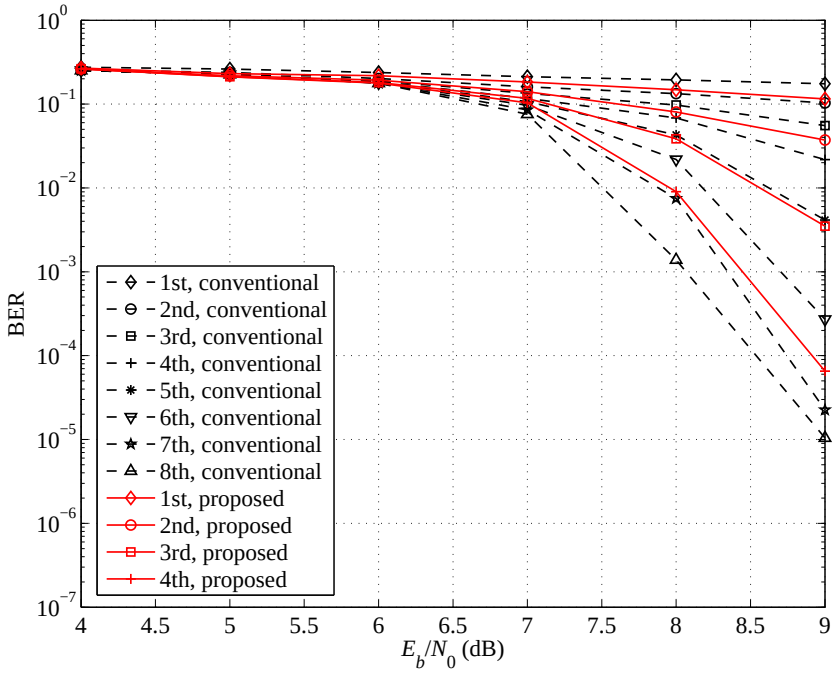


Figure 5.17 BER performance comparison of the conventional and proposed iterative receivers: Sigma mapping and MMSE demodulator.

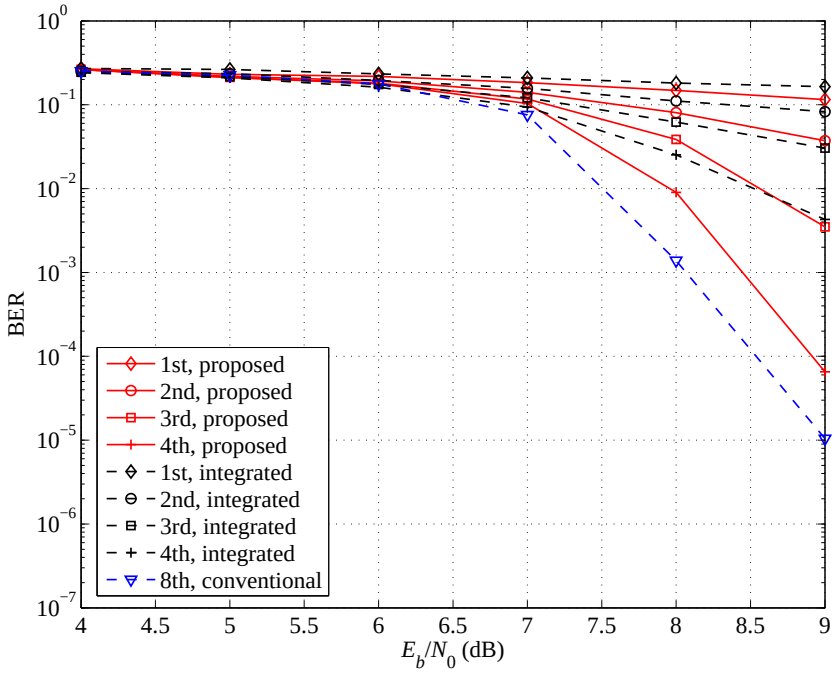


Figure 5.18 BER comparison of different iterative receivers with sigma mapping of 16-QAM.

6. Conclusions and Suggestions for Further Research

6.1 Conclusions

This thesis proposed different iterative receivers for multiuser STBC systems, taking into account two basic properties: (i) The convergence behavior in approaching the asymptotic BER performance and (ii) The overall receiver complexity.

First, the algorithm of the conventional iterative receiver was extended to a more general case with M-QAM constellation in which the symbols of the constellation have different energies. Methods were introduced to quantitatively measure the system complexities with FLOPS and to evaluate the iterative receivers' efficiency with EXIT charts, respectively. These methods provide a foundation to investigate and evaluate different iterative receivers proposed in this thesis. Convenient expressions were developed to simplify the computation of the mutual information between the coded bits and the continuous values of their extrinsic LLR. The EXIT chart technique typically used to investigate two modules' interaction was also extended to the systems with three modules. The resulting visualization of the decoding trajectory makes it much easier to compare different receivers, which have the same asymptotic performance, but different convergence properties.

To overcome the disadvantages of the conventional receiver, namely high complexity and low efficiency, two types of iterative receivers were proposed next. The first receiver is named as INT-MUD-DEM scheme, which is the integrated iterative receiver. It was designed to explore the linearity of sigma mapping. By exploiting

the linear relationship of coded bits and the transmitted channel symbol, the MMSE demodulators and the MMSE-MUD can be combined into one single module. This allows the interference cancellation for multiuser systems to be carried out at the bit-level and therefore, helps to improve the convergence property of the iterative processing. It has been shown that when considering both complexity and convergence property, INT-MUD-DEM scheme is an appropriate candidate to replace the conventional receivers when a few receive antennas and a high-order constellation are employed.

The second iterative receiver is named the two-loop iterative receiver. By introducing an inner iteration loop for the demodulators and the MAP convolutional decoders, besides the outer iteration loop that involves the MUD, the convergence property of the conventional receiver is greatly improved. It was demonstrated that the same asymptotic performance can be approached with fewer iterations (i.e., with a lower computation load).

Furthermore, the question that how many inner loop iterations should be executed per one outer iteration loop was answered with one typical example (systems with 16-QAM and sigma mapping). By investigating the evolution of the extrinsic information transfer characteristic when the number of inner loop iteration increases, it shows that two or three inner-loop iterations per one outer-loop iteration is good enough for systems with 16-QAM and sigma mapping. It is straightforward to apply the proposed principle to determine the best scheme for systems of different configurations.

Finally the effects of phase offsets among the users on multiuser STBC system performance were investigated and discussed. It was demonstrated that, for multiuser coded STBC systems over a Raleigh fading channel, BER performance is not affected by the phase offsets among the users. So all users may employ the same constellation set for simplicity without any loss in BER performance.

6.2 Further Research Topics

This thesis considers the performance of multiuser STBC systems which employ simple two-dimensional constellations. Employing other mapping schemes, such as multi-dimensional constellations/mappings to achieve even better error performance improvement is an attractive subject of further research.

In this thesis, the BER performance comparison of different receiver is carried out with EXIT charts and numerical simulations. However, it is still an interesting topic to develop a more advanced analytical framework for the study of error performance of multiuser STBC systems that use multiple transmit and multiple receive antennas.

Finally, in this thesis, the multiuser STBC systems are only investigated for a frequency non-selective fading channel. It is also of interest to study the applications of multiuser STBC systems over a frequency selective fading channel.

A. A Review of Convolutional Codes

A.1 Encoding of Convolutional Codes

Convolutional codes are popular channel codes since their encoding can be simply implemented with shift registers while their decoding can also be efficiently implemented based on the trellis diagram. The encoder for a convolutional code accepts k-bit blocks of the information $\mathbf{u}$ and produces a coded sequence $\mathbf{v}$ of n-bit blocks. Each encoded block depends not only on the corresponding k-bit message block at the same time unit but also on m previous message blocks. Hence, the encoder has a memory order of m. The ratio $R = k/n$ is called the *code rate*. When $k < n$, or $R < 1$, redundant bits for combating the channel noise are added to the information sequence. Furthermore, more redundancy can be added by increasing the memory order m of the code. How to design the encoder of a convolutional code to achieve reliable transmission over a noisy channel can be found in [13].

Here, an example of a convolutional encoder with $k = 1$, $n = 2$, and $m = 2$ is shown in A.1. Each rectangle in Fig. A.1 represents a shift register. The generator polynomial for the convolutional is given in octal number with the most significant bit denoting the very left (input) connection to the shift register. Therefore, the generator polynomial for this convolutional code can be represented as $G = (5, 7)$ with a code rate $R = 1/2$.

The information sequence $\mathbf{u}$ enters the encoder one bit at a time and two bits $v^{\{0\}}$ and $v^{\{1\}}$ are produced, which make up the coded bits of the output sequence. Because the convolutional encoder is a linear sequential circuit, its operation can be described by a state diagram. The state of an encoder at one time is defined as its

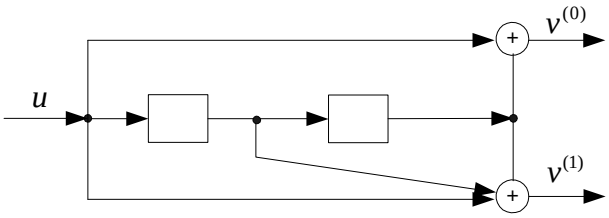


Figure A.1 A rate $R = 1/2$ binary convolutional encoder with $G = (5, 7)$.

shift register contents at that time. For the above code with $G = (5, 7)$, there are four states, denoted as S_0, S_1, S_2, and S_3, respectively. The corresponding contents in the two shift registers are (00), (10), (01), and (11). Note that S_0 always denotes the state with all zeros in the shift registers. The output coded bits at any time instant can be determined according to the input information and the state of the encoder at that time. The determination is usually illustrated with a state diagram, as shown in Fig. A.2.

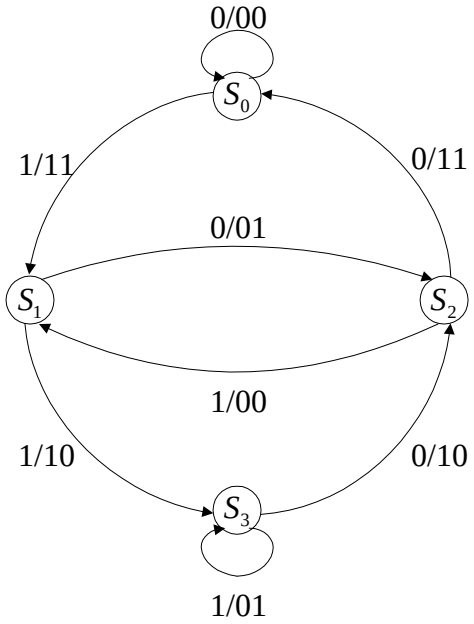


Figure A.2 The state diagram for the encoder with $G = (5, 7)$.

Assuming that the encoder is initially in state S_0, the sequence of the coded bits corresponding to any given information sequence can then be obtained by following

the path through the state diagram determined by the information sequence and noting the corresponding outputs on the branch labels. Note that by adding zero bits to the end of each information block, the encoder can return to state S_0 again. This facilitates the decoding of the information bits at the decoder.

A.2 Decoding of Convolutional Codes

To understand the MAP decoding algorithm, it is helpful to introduce the trellis diagram for the encoder first. The trellis diagram is a result of expanding the state diagram of the encoder in time, that is, representing each time unit with a separate state diagram. Figure A.3 shows the trellis diagram for the encoder with $G = (5, 7)$ and an information sequence of length $h = 5$. The trellis diagram contains 8 time units or levels, and these are labeled from 0 to 7. For a terminated code, assume that the encoder always starts in state S_0 and returns to state S_0. The first $m = 2$ time units correspond to the encoder's departure from state S_0 and the last $m = 2$ time units correspond to the encoder's return to state S_0. It follows that not all states can be reached in the first m or the last m time units for the encoder. However, in the center portion of the trellis, all states are possible, and each time unit contains a replica of the state diagram. There are $2^k = 2$ branches leaving and entering each state. The upper branch leaving each state at time unit i represents the input bit $u_i = 1$, and the lower branch represents $u_i = 0$. Each branch is labeled with the $n = 2$ corresponding outputs $v(i)$. Each path through the trellis represents a unique codeword or information sequence of length $h = 5$. For example, the codeword corresponding to the information sequence $\mathbf{u} = (11101)$ is shown highlighted in Figure A.3.

With the trellis diagram, the MAP decoding algorithm can now be introduced. In the following, a rate-$\frac{1}{n}$ convolutional encoder of overall constraint length ν is considered. The more general case is considered and discussed in [13].

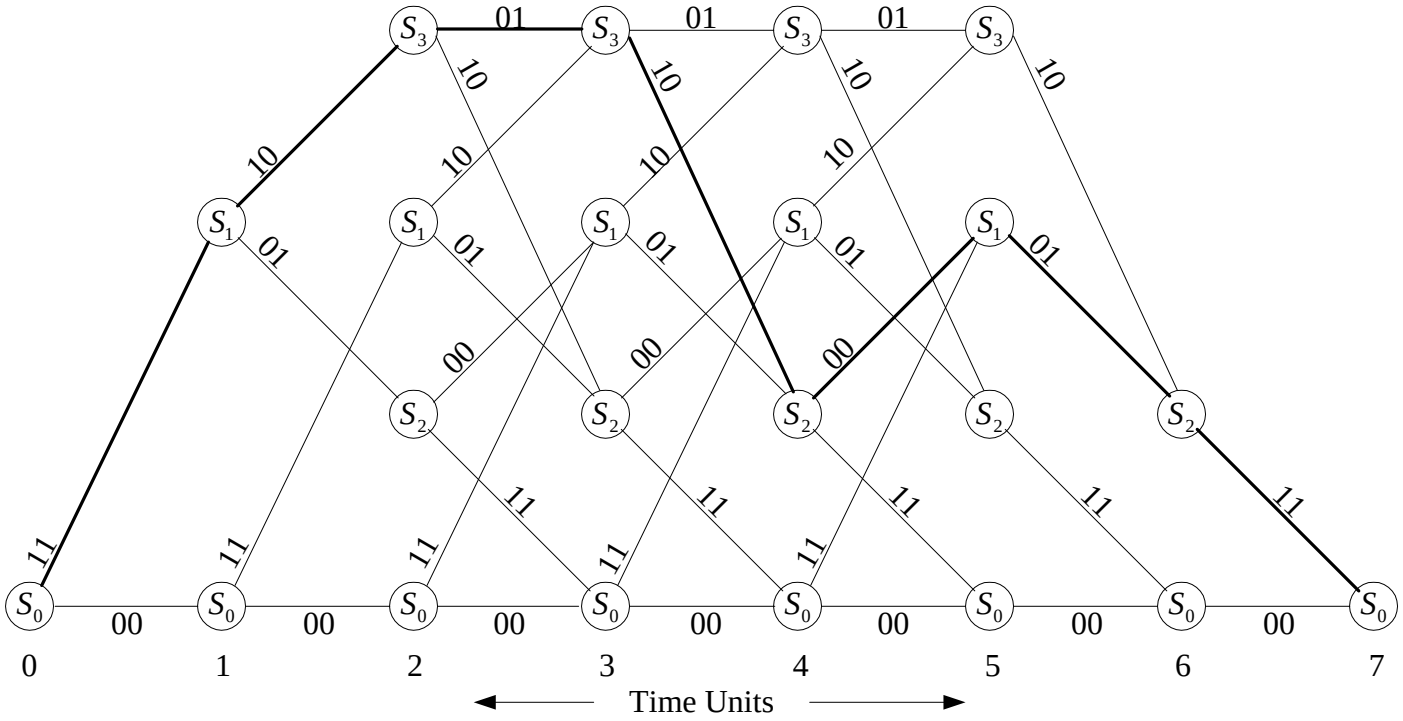


Figure A.3 Trellis diagram for the encoder with $G = (5, 7)$.

First, represent the input block to the encoder at time t by $\underline{d}_t = (d_t^1)$ and the corresponding output of the encoder by $\underline{b}_t = (b_t^1, \cdots, b_t^n)$. Also represent the state of the trellis at time t by a $(\nu-1)$-tuple, as $S_t = (s_t^1, \cdots, s^{(\nu-1)})$. Finally, represent the input information bits that cause the state transition from $S_{t-1} = s'$ to $S_t = s$ by $\underline{d}(s', s)$ and the corresponding output coded bits by $\underline{b}(s', s)$.

Then, define the probability of transition from current time state s' to the next time state s as

$$P[b_t(s', s)] \triangleq P[\underline{b}_t = b_t(s', s)] \tag{A.1}$$

With boundary conditions $\alpha(0) = 1$, $\alpha(s \neq 0) = 0$ at the very first time unit ($t = 0$) and the boundary condotions $\beta_\tau(0) = 1$, $\beta_\tau(s \neq 0) = 0$ at the very last time unit ($t = \tau$), two recursions, known as forward and backward recursions, can be defined as follows:

$$\alpha_t(s) = \sum_{s'} \alpha_{t-1}(s') P[\underline{b}_t(s', s)],\ t = 1, 2, \cdots, \tau \tag{A.2}$$

$$\beta_t(s) = \sum_{s'} \beta_{t+1}(s') P[\underline{b}_{t+1}(s', s)],\ t = \tau - 1, \tau - 2, \cdots, 0 \tag{A.3}$$

where τ is the length of the terminated information sequence. In (A.2) and (A.3) the summations are over all states s' where the transition (s', s) is possible. Probabilities $\alpha_t(s)$ and $\beta_t(s)$ associated with each state at time t reflect the chance that state s is the true state of the encoder at that time unit according to the received signal and the constraints of the code from the forward recursion and backward recursion, respectively.

Let S_j^+ be the set of state pairs (s', s) such that the jth bit of $\underline{b}(s', s)$ is $+1$. Similarly, define S_j^{-1} as the set of state pairs (s', s) such that the jth bit of $\underline{b}(s', s)$ is -1. The *a posteriori* LLR of the coded bit b_t^j at the output of the MAP decoder can

be computed as:

$$\begin{aligned}
\Lambda_2[b_t^j] &\triangleq \log \frac{P[b_t^j = +1|\text{decoding}]}{P[b_t^j = -1|\text{decoding}]} \\
&= \log \frac{\sum_{S_j^+} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s',s)]}{\sum_{S_j^-} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s',s)]} \\
&= \underbrace{\log \frac{\sum_{S_j^+} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i\neq j} P[b_t^i(s',s)]}{\sum_{S_j^-} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i\neq j} P[b_t^i(s',s)]}}_{\lambda_2[b_t^j]} + \underbrace{\log \frac{P[b_t^j = +1]}{P[b_t^j = -1]}}_{\lambda_1^p[b_t^j]} \qquad \text{(A.4)}
\end{aligned}$$

It is seen from (A.4) that the output of MAP decoder is the sum of the *a priori* information $\lambda_1^p[b_t^j]$ and the extrinsic information $\lambda_2[b_t^j]$. The extrinsic information is the information about the coded bit b_t^j gleaned from the *a priori* information about the other coded bits based on the trellis structure of the code.

The *a posteriori* LLR of the information bit can be computed in a similar way. Let $\mathcal{U}_j^+$ be the set of the state pairs (s', s) such that the jth bit of $\underline{d}(s's)$ is $+1$. Similarly, $\mathcal{U}_j^{-1}$ is the set of the state pairs (s', s) such that the jth bit of $\underline{d}(s's)$ is -1. Then,

$$\Lambda_2[d_t^j] = \log \frac{\sum_{\mathcal{U}_j^+} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s',s)]}{\sum_{\mathcal{U}_j^-} \alpha_{t-1}(s') \cdot \beta_t(s) \cdot \prod_{i=1}^{n_0} P[b_t^i(s',s)]} \qquad \text{(A.5)}$$

The information bit d_t^j is then decoded as $\hat{d}_t^j = \text{sgn}(\Lambda_2[d_t^j])$.

B. A Review of Mutual Information

B.1 Definition of the Mutual Information and its Properties.

Mutual information is a basic concept in information theory. It is a measure of the amount of information that one random variable contains about another random variable. It measures the reduction in the uncertainty of one random variable due to the knowledge of the other. Specifically, given two random variables X and Y, the mutual information $I(X;Y)$ is defined as follows:

$$I(X;Y) = H(X) - H(X|Y) \tag{B.1}$$

where $H(\cdot)$ is the entropy of a random variable which measures the uncertainty associated with it. For a continuous random variable X, $H(X)$ is defined as

$$H(X) = -\int p(x)\log_2 p(x)dx \tag{B.2}$$

If X is a discrete random variable, $H(X)$ is defined as follows:

$$H(X) = -\sum p(x)\log_2 p(x)dx \tag{B.3}$$

In both cases $p(X)$ represents the marginal probability distribution of random variable X. Based on the above equations, it is apparent why the entropy is often considered a measure of uncertainty. As an example, let X represent the output information bits from one specific source. If there is no uncertainty about the output bits, i.e., they are always 1 ($p(X=1)=1, p(X=0)=0$) or always be 0 ($p(X=1)=0, p(X=0)=1$), then the entropy $H(X)$ equals zero. If however, there is a high uncertainty about the output bits, which can be 1 or 0 equally likely ($p(X=1)=0.5, p(X=0)=0.5$), then the entropy $H(X)$ equals 1.

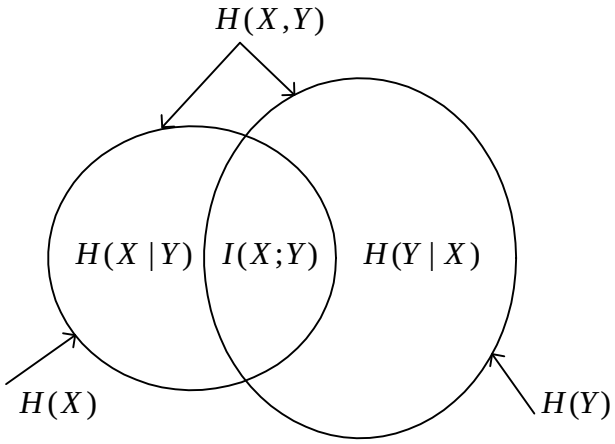


Figure B.1 Relationship between entropy and mutual information.

Using the Bayes' rule on conditional probabilities, (B.1) can be rewritten as

$$I(X;Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y) \tag{B.4}$$

The relationship among $H(X)$, $H(Y)$, $H(X,Y)$, $H(X|Y)$,$H(Y|X)$ and $I(X;Y)$ is illustrated with a Venn diagram in Fig. B.1 [21].

Observe that the mutual information $I(X;Y)$ corresponds to the intersection of the information in X with the information in Y. Intuitively, if X and Y are independent, two circles in Fig. B.1 are completely separated. Then X contains no information about Y and knowing X does not give any information about Y, so their mutual information is zero ($I(X;Y) = 0$). On the other hand, if X and Y are identical, two circles in Fig. B.1 are exactly overlapped. Therefore the mutual information is the same as the entropy of Y or X ($I(X;Y) = H(X) = H(Y)$), and knowing X provides all the necessary information about Y and vice versa.

B.2 Computation of the Mutual Information

With a simple derivation, the mutual information of two discrete random variables X and Y can be represented as [21]:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{B.5}$$

where $p(x, y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively.

In the continuous case, the summation is replaced by a definite integral, hence

$$I(X;Y) = \int_y \int_x p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} dxdy \tag{B.6}$$

where $p(x, y)$ is the joint probability density function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y respectively.

Accordingly, for the channel with discrete inputs $X = \{x_k\}, 1 \leq k \leq M$ and continuous outputs $R = \{r\}$, the mutual information between X and R is

$$I(X;R) = \sum_{k=1}^{M} \int_{-\infty}^{\infty} p(r, x_k) \log_2 \frac{p(r, x_k)}{p(r)p(x_k)} dr \tag{B.7}$$

where $p(r, x_k)$, $p(r)$ and $p(x_k)$ are defined similarly as above. Using the Bayes' rule on conditional probabilities, (B.7) can be rewritten as

$$I(X;R) = \sum_{k=1}^{M} p(x_k) \int_{-\infty}^{\infty} p(r|x_k) \log_2 \frac{p(r|x_k)}{p(r)} dr \tag{B.8}$$

For the special case of binary symmetric channel (BSC) with $M = 2$, and when the input bits are equally likely ($p(X = 0) = p(X = 1) = 0.5$), the mutual information $I(X; R)$ can be computed as

$$I(X;R) = \frac{1}{2} \sum_{x=1,-1} \int_{-\infty}^{\infty} p(r|X = x) \log_2 \frac{2p(r|X = x)}{p(r|X = 1) + p(r|X = -1)} dr \tag{B.9}$$

C. MMSE Demodulator and its Complexity

C.1 Soft Instantaneous MMSE Interference Cancellation

A general discrete-time system is given here to demonstrate how the Wang-Poor algorithm can be effectively used for demodulation when sigma mapping is employed. The system's input/output equation can be defined as

$$\underline{y}(i) = \underline{\underline{H}}\underline{b}(i) + \underline{n}(i) \tag{C.1}$$

where $\underline{y}(i)$ is the received vector at time i, which consists of n real signal components, $\underline{\underline{H}}$ is the equivalent channel response matrix, $\underline{b}(i) \triangleq [\underline{b}_1(i), \underline{b}_2(i), \cdots, \underline{b}_K(i)]^T$ is the K-vector of the transmitted coded bits at time i, and $\underline{n}(i)$ is the vector of i.i.d. Gaussian noise samples with covariance matrix $\sigma^2\underline{\underline{R}}$ at time i.

The matrix $\underline{\underline{H}}$ is not always a square matrix, and for a higher constellation, n is always less than K. One way to convert the model in (C.1) to an appropriate form is by multiplying $\underline{\underline{H}}^T$ on both sides to obtain

$$\underline{\underline{H}}^T\underline{y}(i) = \underline{\underline{H}}^T\underline{\underline{H}}\underline{b}(i) + \underline{\underline{H}}^T\underline{n}(i) \tag{C.2}$$

Then the Wang-Poor algorithm can be applied directly to solve C.2. However this is not the optimal way in terms of the computation complexity. In order to have a lower complexity, we derive the MMSE demodulator directly based on the system in (C.1), rather than using the equivalent form in (C.2).

Assume that the *a priori* probability of the coded bits $P[b_k(i)]$ is available from the previous stage (i.e., the SISO channel decoder). The soft estimates of the coded

bits can be formed as

$$\tilde{b}_k(i) \triangleq E\{b_k(i)\} = \sum_{b_j \in \{+1,-1\}} b_j P[b_k(i) = b_j] \tag{C.3}$$

Define

$$\begin{aligned} \underline{\tilde{b}}(i) &= [\tilde{b}_1(i), \cdots, \tilde{b}_K(i)] && \text{(C.4)} \\ \underline{\tilde{b}}_k(i) &\triangleq \underline{\tilde{b}}(i) - \tilde{b}_k(i)\underline{e}_k \\ &= [\tilde{b}_1(i), \cdots, \tilde{b}_{k-1}(i), 0, \tilde{b}_{k+1}(i), \cdots, \tilde{b}_K(i)]^T && \text{(C.5)} \end{aligned}$$

where $\underline{e}_k$ denotes a K-vector of all zeros, except for the kth element, which is 1. Therefore $\underline{\tilde{b}}_k(i)$ is obtained from $\underline{\tilde{b}}(i)$ by setting the kth element to zero. For each coded bit, a soft interference cancellation is performed on the received signal $\underline{y}(i)$ in (C.1), to obtain

$$\underline{\tilde{y}}_k(i) \triangleq \underline{y}(i) - \underline{H} \cdot \underline{\tilde{b}}_k(i), \ k = 1, \cdots, K \tag{C.6}$$

Next, in order to further suppress the residual interference in $\underline{\tilde{y}}_k(i)$, an instantaneous linear MMSE filter [11] $\underline{w}_k(i)$ is applied to $\underline{\tilde{y}}_k(i)$ to obtain

$$\hat{b}_k(i) = \underline{w}_k^T(i) \cdot \underline{\tilde{y}}_k(i) \tag{C.7}$$

where the filter $\underline{w}_k(i) \in \mathbb{R}^K$ is chosen to minimize the mean-square error between the coded bits $b_k(i)$ and the filter output $\hat{b}_k(i)$, i.e.,

$$\underline{w}_k(i) = \arg \min_{\underline{w} \in \mathbb{R}^K} E\{[b_k(i) - \underline{w}^T \underline{y}_k(i)]^2\} \tag{C.8}$$

With a simple derivation, one can obtain

$$\underline{w}_k(i) = E\{\underline{\tilde{y}}_k(i)\underline{\tilde{y}}_k^T(i)\}^{-1} \cdot E\{b_k(i)\underline{\tilde{y}}_k(i)\} \tag{C.9}$$

The derivations of $E\{\underline{\tilde{y}}_k(i)\underline{\tilde{y}}_k^T(i)\}$ and $E\{b_k(i)\underline{\tilde{y}}_k(i)\}$ are given next. First, due to the effect of random interleaver, all the transmitted bits can be assumed to be

independent. Thus

$$E\{b_k(i)[\underline{b}(i)-\underline{\tilde{b}}_k(i)]\} = E\left\{ b_k(i)\cdot \begin{bmatrix} b_1(i)-\tilde{b}_1(i) \\ \vdots \\ b_{k-1}(i)-\tilde{b}_{k-1}(i) \\ b_k(i) \\ b_{k+1}(i)-\tilde{b}_{k+1}(i) \\ \vdots \\ b_K(i)-\tilde{b}_K(i) \end{bmatrix} \right\}$$

$$= \begin{bmatrix} E\{b_k(i)\}\cdot E\{[b_1(i)-\tilde{b}_1(i)]\} \\ \vdots \\ E\{b_k(i)\}\cdot E\{[b_{k-1}(i)-\tilde{b}_{k-1}(i)]\} \\ E\{b_k(i)\cdot b_k(i)\} \\ E\{b_k(i)\}\cdot E\{[b_{k+1}(i)-\tilde{b}_{k+1}(i)]\} \\ \vdots \\ E\{b_k(i)\}\cdot E\{[b_K(i)-\tilde{b}_K(i)]\} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \underline{e}_k \qquad \text{(C.10)}$$

From (C.1) and (C.6), one has

$$\underline{\tilde{y}}_k(i) = \underline{y}(i) - \underline{H}\cdot\underline{\tilde{b}}_k(i) = \underline{H}[\underline{b}(i)-\underline{\tilde{b}}_k(i)] + \underline{n} \qquad \text{(C.11)}$$

It follows from (C.10) and (C.11) that

$$\begin{aligned} E\{\underline{\tilde{y}}_k(i)\underline{\tilde{y}}_k^T(i)\} &= \underline{H}\text{cov}\{\underline{b}(i)-\underline{\tilde{b}}_k(i)\}\underline{H}^T + \sigma^2\underline{R} \\ &= \underline{H}V_k(i)\underline{H}^T + \sigma^2\underline{R} \end{aligned} \qquad \text{(C.12)}$$

where

$$\begin{aligned} V_k(i) &\triangleq \text{cov}\{\underline{b}(i)-\underline{\tilde{b}}_k(i)\} = \text{diag}\ [\text{var}\{b_1(i)\},\cdots,\text{var}\{b_{k-1}(i)\},1, \\ &\qquad\qquad \text{var}\{b_{k+1}(i)\},\cdots,\text{var}\{b_K(i)\}] \\ &= \text{diag}\ [1-\tilde{b}_1^2(i),\cdots,1-\tilde{b}_{k-1}^2(i),1,\ \ 1-\tilde{b}_{k+1}^2(i),\cdots,1-\tilde{b}_K^2(i)] \end{aligned} \qquad \text{(C.13)}$$

and

$$E\{b_k(i)\underline{\tilde{y}}_k(i)\} = \underline{H}E\{b_k(i)(\underline{b}(i)-\tilde{b}_k(i))\} + \underbrace{E\{\underline{n}\}}_{=0}E\{b_k(i)\} = \underline{H}\underline{e}_k \qquad \text{(C.14)}$$

Substituting (C.12) and (C.14) into (C.9) and (C.7), $\underline{w}_k(i)$ and the soft estimates of coded bit $b_k(i)$ can be obtained, respectively, as

$$\underline{w}_k(i) = [\underline{H}V_k(i)\underline{H}^T + \sigma^2\underline{R}]^{-1}\underline{H}e_k \tag{C.15}$$

$$\hat{b}_k(i) = \underline{e}_k^T\underline{H}^T[\underline{H}V_k(i)\underline{H}^T + \sigma^2\underline{R}]^{-1}\tilde{\underline{y}}_k(i) \tag{C.16}$$

The instantaneous MMSE filtering used here provides an efficient and accurate way to compute the extrinsic information, which is vital to the iterative multiuser receiver.

C.2 Gaussian Approximation of the Soft MMSE Filter's Output

It is shown that the distribution of the residual interference-plus-noise at the output of a linear MMSE multiuser detector is well approximated by a Gaussian distribution [33]. So it can be assumed that the output of the soft instantaneous MMSE filter $\hat{b}_k(i)$ can be represented as the output of an equivalent additive white Gaussian noise channel having $b_k(i)$ as its input [11]:

$$\hat{b}_k(i) = \mu_k(i)b_k(i) + \eta_k(i) \tag{C.17}$$

where $\mu_k(i)$ is the equivalent amplitude of the kth coded bit, and $\eta_k(i) \sim \mathcal{N}(0, \nu_k^2(i))$ is a Gaussian noise sample. The parameters $\mu_k(i)$ and $\nu_k^2(i)$ can be computed as follows, where the expectation is taken with respect to the interfering coded bits $\{b_j(i)\}_{j\neq k}$ and the channel noise $\underline{n}(i)$:

$$\mu_k(i) = E\{\hat{b}_k(i)b_k(i)\} \tag{C.18}$$

$$\nu_k^2(i) = \mathrm{var}\{\hat{b}_k(i)\} = E\{\hat{b}_k^2(i)\} - \mu_k^2(i) \tag{C.19}$$

The derivations of $\mu_k(i)$ and $\nu_k^2(i)$ are given next. First, multiply both sides of (C.17) by $b_k(i)$ and compute its expectation as:

$$\begin{aligned} E\{\hat{b}_k(i)b_k(i)\} &= E\{(\mu_k(i)b_k(i) + \eta_k(i)) \cdot b_k(i)\} \\ &= \mu_k(i)\underbrace{E\{b_k(i)^2\}}_{=1} + \underbrace{E\{\eta_k(i)\}}_{=0}E\{b_k(i)\} = \mu_k(i) \end{aligned} \tag{C.20}$$

Next, multiply (C.16) by $b_k(i)$, and substitute in (C.14) to compute

$$\begin{aligned} E\{\hat{b}_k(i)b_k(i)\} &= E\{\underline{e}_k^T\underline{H}^T[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}\underline{\tilde{y}}_k(i)\cdot b_k(i)\} \\ &= \underline{e}_k^T\underline{H}^T[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}E\{\underline{\tilde{y}}_k(i)\cdot b_k(i)\} \\ &= \underline{e}_k^T\underline{H}^T[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}\underline{H}\underline{e}_k \end{aligned} \tag{C.21}$$

Comparing (C.21) and (C.20), one obtains

$$\begin{aligned} \mu_k(i) &= \underline{e}_k^T\underline{H}^T[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}\underline{H}\underline{e}_k \\ &= \{\underline{H}^T[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}\underline{H}\}_{kk} \end{aligned} \tag{C.22}$$

Now substitute (C.7), (C.12) and (C.15) into (C.19), $\nu_k^2(i)$ can be obtained as

$$\begin{aligned} \nu_k^2(i) &= E\{\hat{b}_k^2(i)\}-\mu_k^2(i) = E\{[\underline{w}_k^T(i)\cdot\underline{\tilde{y}}_k(i)]^2\}-\mu_k^2(i) \qquad \text{[with (C.7)]} \\ &= E\{[\underline{w}_k^T(i)\cdot\underline{\tilde{y}}_k(i)][\underline{\tilde{y}}_k^T(i)\underline{w}_k^T(i)]\}-\mu_k^2(i) = \underline{w}_k^T(i)E\{\underline{\tilde{y}}_k(i)\underline{\tilde{y}}_k^T(i)\}\underline{w}_k^T(i)-\mu_k^2(i) \\ &= \{[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}\underline{H}\underline{e}_k\}^T\cdot\{\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}\} \\ &\qquad \cdot\{[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}\underline{H}\underline{e}_k\}-\mu_k^2(i) \quad \text{[with (C.12) and (C.15)]} \\ &= \underline{e}_k^T\underline{H}^T[\underline{H}V_k(i)\underline{H}^T+\sigma^2\underline{R}]^{-1}\underline{H}\underline{e}_k-\mu_k^2(i) \\ &= \mu_k(i)-\mu_k^2(i) \qquad \text{[with (C.22)]} \end{aligned} \tag{C.23}$$

From (C.17), (C.22) and (C.23), the *a posteriori* probability of the coded bits delivered by the soft instantaneous MMSE filter is

$$p[\hat{b}_k(i)|b_k(i)=\pm 1] = \sqrt{\frac{1}{2\pi E\{\nu_k^2(i)\}}}\cdot\exp\left(-\frac{\|\hat{b}_k(i)\mp\mu_k(i)\|^2}{2E\{\nu_k^2(i)\}}\right) \tag{C.24}$$

The extrinsic information can be obtained accordingly as

$$\lambda_1[b_k(i)] \triangleq \log\frac{p[\hat{b}_k(i)|b_k(i)=+1]}{p[\hat{b}_k(i)|b_k(i)=-1]} = \frac{2\hat{b}_k(i)}{1-\mu_k(i)} \tag{C.25}$$

which then can be fed back to the SISO channel decoder for further iteration.

C.3 Complexity of the MMSE Demodulator

The MMSE demodulator estimates the coded bits carried by each transmitted symbol based on the information passed from the MMSE detector (symbol estimates)

and from the channel decoder (the *a priori* probabilities of the coded bits). The complexity of the MMSE demodulator is first evaluated based on one symbol and then converted against each coded bit. The steps to determine the complexity are summarized as follows:

1. Compute $\tilde{b}_k(i)$ for all K coded bits ($k = 1, \cdots, K$) as:

$$\tilde{b}_k(i) \triangleq E\{b_k(i)\} = \sum_{b_j \in \{+1,-1\}} b_j P[b_k(i) = b_j] \qquad \text{[This is Eqn. (C.3)]}$$

Note that the *a priori* probability $P[b_k(i) = b_j]$ is provided by MAP channel decoder. Since b_j is ± 1, the computation in (C.3) actually only involves addition/substraction and no multiplication/division. So the computation load for $\tilde{b}_k(i)$ can be ignored, i.e.,

$$L_1 = 0 \text{ (MUL)} \tag{C.26}$$

2. Compute $\underline{\tilde{y}}_k(i)$ for all K coded bits as:

$$\begin{aligned} \underline{\tilde{y}}_k(i) &\triangleq \underline{y}(i) - \underline{H} \cdot \underline{\tilde{b}}_k(i) = \underline{y}(i) - \underline{H}[\underline{\tilde{b}}(i) - \tilde{b}_k(i)\underline{e}_k] \\ &= \underbrace{\underline{y}(i) - \underline{H}\underline{\tilde{b}}(i)}_{\text{Part 1}} + \underbrace{\underline{H}\tilde{b}_k(i)\underline{e}_k}_{\text{Part 2}} \end{aligned} \tag{C.27}$$

Observe that Part 1 is common for all K coded bits and needs to be computed only once. The calculation of this part needs nK (MULs); The computation of Part 2 is more special and needs to be done separately for each coded bit with n (MULs). Therefore for all K coded bits, the total computation load for $\underline{\tilde{y}}_k(i)$ is

$$L_2 = \underbrace{nK}_{\text{Part 1}} + \underbrace{n \times K}_{\text{Part 2}} = 2nK(\text{MULs}) \tag{C.28}$$

3. Compute $\hat{b}_k(i)$ for all K coded bits ($k = 1, \cdots, K$) as:

$$\hat{b}_k(i) = \underline{e}_k^T \underline{H}^T [\underline{H} V_k(i) \underline{H}^T + \sigma^2 \underline{R}]^{-1} \underline{\tilde{y}}_k(i) \qquad \text{[This is Eqn. (C.16)]}$$

The computation of $\hat{b}_k(i)$ could be divided into the following five sub-steps:

(a) First, it takes K(MULs) to compute $\|\tilde{b}_k(i)\|^2, k = 1, \cdots, K$.

(b) In order to have an efficient calculation of $\mathbf{H}V_k(i)\mathbf{H}^H$, define

$$\begin{aligned}\mathbf{V} &= \text{diag } [1-\tilde{b}_1^2(i),\cdots,1-\tilde{b}_{k-1}^2(i),1-\tilde{b}_k^2(i),\\ &\qquad 1-\tilde{b}_{k+1}^2(i),\cdots,1-\tilde{b}_K^2(i)] \end{aligned} \tag{C.29}$$

Then

$$\begin{aligned}\mathbf{H}\mathbf{V}_k(i)\mathbf{H}^H &= \mathbf{H}\left[\mathbf{V}+\text{diag}\{0,\cdots,0,\|\tilde{b}_k(i)\|^2,0,\cdots,0\}\right]\mathbf{H}^H\\ &= \underbrace{\mathbf{H}\mathbf{V}\mathbf{H}^H}_{\text{Part 1}}+\underbrace{[\mathbf{H}\mathbf{e}_k(i)]\|\tilde{b}_k(i)\|^2[\mathbf{H}\mathbf{e}_k(i)]^T}_{\text{Part 2}} \end{aligned} \tag{C.30}$$

As before, Part 1 is common for all K coded bits and it needs to be determined once per block. Because $\mathbf{V}$ is a real diagonal matrix and $\mathbf{H}$ is a real matrix, it takes $K\cdot n$ (MULs) to compute $\mathbf{V}\mathbf{H}^H$; And it takes $K\cdot n^2$ (MULs) to compute $\mathbf{H}\mathbf{V}\mathbf{H}^H$. Therefore calculating Part 1 requires a total of $Kn+Kn^2$ (MULs). Next, Part 2 needs to be computed separately for each coded bit. For one bit, n (MULs) are required to compute $\|\tilde{b}_k(i)\|^2[\mathbf{H}\mathbf{e}_k(i)]^T$ and n^2 (MULs) are required to compute $[\mathbf{H}\mathbf{e}_k(i)]\|\tilde{b}_k(i)\|^2[\mathbf{H}\mathbf{e}_k(i)]^T$. Therefore, to calculate Part 2 a total of $K[n^2+n]$ (MULs) are needed.

Combining the computations of Part 1 and Part 2, it takes $2K(n+n^2)$ (MULs) to compute $\mathbf{H}\mathbf{V}_k(i)\mathbf{H}^H$. Furthermore, given the symmetry property of the matrix, almost half of the computation load could be saved. The computation load is counted as $K(n^2+n)$ (MULs).

(c) Compute $\mathbf{T}_1 \triangleq [\mathbf{H}\mathbf{V}_k(i)\mathbf{H}^H+\sigma^2\underline{R}]^{-1}$. This takes $n^3/3$ (MULs) [14] for one coded bit $\hat{b}_k(i)$ to do the matrix inversion. Therefore for all K coded bits, it requires $\frac{K}{3}n^3$ (MULs).

(d) Compute $\mathbf{T}_2 \triangleq \tilde{\mathbf{e}}_k^H(i)\cdot\mathbf{H}^H\cdot\mathbf{T}_1 = \tilde{\mathbf{e}}_k^H(i)\cdot\mathbf{H}^H\cdot[\mathbf{H}\mathbf{V}_k(i)\mathbf{H}^H+\sigma^2\underline{R}]^{-1}$. Due to the effect of $\tilde{\mathbf{e}}_k^H(i)$, one only needs to compute one row of matrix $\mathbf{H}^H\cdot\mathbf{T}_1$, which takes n^2 (MULs). So for all K coded bits, it requires Kn^2 (MULs) to compute $\mathbf{T}_2$.

(e) Compute $\mathbf{T}_3 \triangleq \mathbf{T}_2\cdot\underline{\tilde{y}}_k(i)$. For one coded bits, it requires n (MULs) to compute $\mathbf{T}_3$. So for all K coded bits it requires nK (MULs).

In summary, in order to compute $\hat{b}_k(i)$ in (C.16), the computation load is

$$\begin{aligned} L_3 &= \underbrace{K}_{\text{(a)}} + \underbrace{2K(n^2+n)}_{\text{(b)}} + \underbrace{Kn^3/3}_{\text{(c)}} + \underbrace{Kn^2}_{\text{(d)}} + \underbrace{Kn}_{\text{(e)}} \text{ (MULs)} \\ &= \frac{K}{3}n^3 + 3Kn^2 + 3Kn + K \text{ (MULs)} \end{aligned} \tag{C.31}$$

4. Compute $\mu_k(i)$ as follows:

$$\begin{aligned} \mu_k(l) &\triangleq \{\mathbf{H}^H[\mathbf{H}\mathbf{V}_k(i)\mathbf{H}^H + \sigma^2\underline{R}]^{-1}\mathbf{H}\}_{kk} \qquad \text{[This is Eqn. (C.22)]} \\ &= \tilde{\mathbf{e}}_k^H(i) \cdot \mathbf{H}^H[\mathbf{H}\mathbf{V}_k(i)\mathbf{H}^H + \sigma^2\underline{R}]^{-1}\mathbf{H} \cdot \tilde{\mathbf{e}}_k(i) \\ &= \mathbf{T}_2 \cdot \{\mathbf{H} \cdot \tilde{\mathbf{e}}_k(i)\} \end{aligned} \tag{C.32}$$

Since $\mathbf{T}_2$ is already available from the previous calculation, it takes n (MULs) to calculate the inner product of $\mathbf{T}_2 \cdot \{\mathbf{H} \cdot \tilde{\mathbf{e}}_k(i)\}$. So the total computation load to compute $\mu_k(i)$ for all K coded bits is:

$$L_4 = Kn \text{ (MULs)} \tag{C.33}$$

5. Compute $E\{\nu_k^2(i)\}$ as:

$$E\{\nu_k^2(l)\} = \mu_k(l) - \mu_k^2(l) \qquad \text{[This is Eqn. (C.23)]}$$

It can be easily seen that for K coded bits, the total computation load is:

$$L_5 = K \text{ (MULs)} \tag{C.34}$$

6. Compute $\lambda_1[b_k(i)]$ as:

$$\lambda_1[b_k(i)] = \frac{2\hat{b}_k(i)}{1-\mu_k(i)} \qquad \text{[This is Eqn. (C.25)]}$$

One multiplication (1 MUL) and one division (1 MUL) are involved in the calculation of $\lambda_1[b_k(i)]$. Thus, the total computation load to compute $\lambda_1[b_k(i)]$ for K coded bits is:

$$L_6 = 2K \text{ (MULs)} \tag{C.35}$$

Combining all the computation loads given in (C.26), (C.28), (C.31), (C.33), (C.34) and (C.35), the total complexity of the MMSE demodulator per K coded bits is:

$$\begin{aligned} L_{\text{total}}^{\text{MMSE-DEMO}} &= \underbrace{0}_{L_1} + \underbrace{2Kn}_{L_2} + \underbrace{Kn^3/3 + 3Kn^2 + 3Kn + K}_{L_3} + \underbrace{Kn}_{L_4} + \underbrace{K}_{L_5} \underbrace{2K}_{L_6} \\ &= Kn^3/3 + 3Kn^2 + 6Kn + 4K \text{ (MULs)} \end{aligned}$$

Finally, the computation load per one coded bit for the MMSE demodulator is expressed as

$$L^{\text{MMSE-DEMO}} = \frac{L_{\text{total}}^{\text{MMSE-DEMO}}}{K} = n^3/3 + 3n^2 + 6n + 4 \text{ (MULs)} \qquad \text{(C.36)}$$

References

[1] Y. Jing, "Space-time code design and its applications in wireless networks," *Ph.D. thesis, California Institute of Technology Pasadena, California, USA*, Aug. 2004.

[2] M. S. Alouini and A. J. Goldsmith, "Capacity of Rayleigh fading channels under different adaptive transmission and diversity-combining techniques," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 1165–1181, July 1999.

[3] W. C. Jakes, *Microwave Mobile Communication*. Piscataway, NJ: IEEE Press, 2nd ed., 1994.

[4] R. S. Blum, Y. G. Li, J. H. Winters, and Q. Yan, "Improved space-time coding for MIMO-OFDM wireless communications," *IEEE Trans. Commun.*, vol. 49, pp. 1873–1878, Nov. 2001.

[5] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, New York, 1st ed., 2005.

[6] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas in Commun.*, vol. 16, pp. 1451–1458, Oct. 1998.

[7] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, p. 744765, Mar. 1998.

[8] V. Tarokh, H. Jafarkhani, and A. Calderbank, "Space-time block codes for orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1456–1467, July 1999.

[9] B. Lu and X. Wang, "Iterative receivers for multiuser space-time coding systems," *IEEE J. Select. Areas in Commun.*, vol. 18, pp. 2322–2335, Nov. 2000.

[10] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85–96, Jan. 1986.

[11] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancelation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 1046–1061, July 1999.

[12] A. Matache, C. Jones, and R. Wesel, "Reduced complexity MIMO detectors for LDPC coded systems," *IEEE Military Commun. Conference*, pp. 1073–1079, 2004.

[13] S. Lin and D. J. Constello, *Error Control Coding*. Upper Saddle River, New Jersey, 2nd ed., 2005.

[14] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Press, 3rd ed., 1996.

[15] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[16] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[17] H. E. Gamal and A. Hammons, "Analyzing the turbo decoder using the Gaussian approxiamtion," *IEEE J. Select. Areas in Commun.*, vol. 47, pp. 671–886, Feb. 2001.

[18] S. ten Brink, J. Speidel, and R. Yan, "Iterative demapping and decoding for multilevel modulation," *Proc. IEEE GLOBECOM*, pp. 579–584, Nov. 1998.

[19] B. Scanavino, G. Mondorsi, and S. Benedetto, "Convergence properties of iterative decoders working at bit and symbol level," *Proc. IEEE GLOBECOM*, pp. 1037–1041, Nov. 2001.

[20] M. Tüchler, S. ten Brink, and J. Hagenauer, "Measures for tracing convergence of iterative decoding algorithms," *Proc. 4th IEEE/ITG Conf. Source, channel coding, Berlin, Germany*, vol. 47, pp. 53–60, Jan. 2002.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley, 1991.

[22] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and covolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[23] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEU Int. J. Electron. Commun.*, vol. 54, pp. 389–398, Dec. 2000.

[24] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Trans. Commun.*, vol. 52, pp. 209–218, Feb. 2004.

[25] N. H. Tran, H. H. Nguyen, and T. Le-Ngoc, "Performance of BICM-ID with signal space diversity," to appear in *IEEE Trans. Wireless Commu.*

[26] L. Duan, B. Rimoldi, and R. Urbanke, "Approaching the AWGN channel capacity without active shaping," *Proc. IEEE Int. Symp. Information Theory, Ulm, Germany*, p. 374, June 1997.

[27] X. Ma and L. Ping, "Coded modulation using superimposed binary codes," *IEEE Trans. Inform. Theory*, pp. 3331–3343, Dec. 2004.

[28] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcing and decoding: Turbo-codeds," *Proc. IEEE Int. Conf. Communications, Geneva, Switzerland*, pp. 1064–1070, May 1993.

[29] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asentorfer, "Iterative multiuser detection for CDMA with FEC: near-single-user performance," *IEEE Trans. Commun.*, vol. 46, pp. 1693–1699, Dec. 1998.

[30] Z. Shi and C. Schlegel, "Joint iterative decoding of serially concatenated error control coded CDMA," *IEEE J. Select. Areas in Commun.*, vol. 19, pp. 1646–1653, Aug. 2001.

[31] H. H. Nguyen, H. G. Vu, and D. E. Dodds, "Multiple access systems with QPSK modulation," *IEICE Transactions on Fundamentals*, vol. E87-A, pp. 1833–1835, July 2004.

[32] B. M. Hochwal and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, pp. 389–399, Mar. 2003.

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