國 立 交 通 大 學

電子工程學系電子研究所

博 士 論 文

多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究

Research on Sphere/LDPC Decoder for Coded MIMO Systems

研 究 生 ：廖彥欽

指導教授 ：張錫嘉

劉志尉

中華民國 九十七 年 二 月

多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究

Research on Sphere/LDPC Decoder for Coded MIMO Systems

研 究 生： 廖彥欽　　Student： Yen-Chin Liao

指導教授： 張錫嘉 博士　　Advisor： Dr. Hsie-Chia Chang

劉志尉 博士　　Dr. Chih-Wei Liu

國 立 交 通 大 學

電子工程學系電子研究所

博 士 論 文

A Dissertation

Submitted to Department of Electronics Engineering & Institute Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

in

Electronics Engineering

February 2008

Hsinchu, Taiwan, Republic of China.

中華民國 九十七 年 二 月

國 立 交 通 大 學

博碩士論文全文電子檔著作權授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學電子工程系所
系統 組， 96 學年度第 二 學期取得博士學位之論文。

論文題目：多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究
指導教授：張錫嘉、劉志尉

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：

本校及台灣聯合大學系統區域網路	■ 中華民國 98 年 2 月 14 日公開
校外網際網路	■ 中華民國 98 年 2 月 14 日公開

■ **全文電子檔送交國家圖書館**

授 權 人：廖彥欽

親筆簽名： 廖彥欽

中華民國 97 **年** 2 **月** 14 **日**

國 立 交 通 大 學

博碩士紙本論文著作權授權書

（提供授權人裝訂於全文電子檔授權書之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學電子工程系所系統組，96 學年度第二學期取得博士學位之論文。

論文題目：多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究
指導教授：張錫嘉、劉志尉

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：＿＿＿＿＿＿＿＿＿＿＿＿，請將論文延至＿＿＿年＿＿＿月＿＿＿日再公開。

授 權 人：廖彥欽

親筆簽名： 廖彥欽

中華民國 97 **年** 2 **月** 14 **日**

國家圖書館
博碩士論文電子檔案上網授權書

（提供授權人裝訂於紙本論文本校授權書之後）

ID:GT009211826

本授權書所授權之論文為授權人在國立交通大學電子工程系所 96 學年度第 二 學期取得博士學位之論文。

論文題目：多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究
指導教授：張錫嘉、劉志尉

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

※ 讀者基於非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：廖彥欽

親筆簽名： 廖彥欽

民國 97 **年** 2 **月** 14 **日**

推 薦 函

主旨：推薦電子工程學系博士班研究生廖彥欽，參加國立交通大學博士學位口試。

說明：本人所指導之博士班學生廖彥欽，業已通過資格考試，並完成本校電子工程學系電子研究所博士班規定之學科課程及論文研究訓練。廖同學主要從事先進通訊系統與通道解碼器之研究工作，其論文題目「多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究」（Research on Sphere/LDPC Decoder for Coded-MIMO Systems）針對多輸入輸出系統提出高效能與低成本解碼方式，透過理論分析與推導以及模擬實驗，完成各式高效能演算法與架構。此外，廖同學於先進通道解碼器亦多所著墨，以理論分析改善實作上之困難與誤差，並針對不同系統提出一系列演算法與架構，以實驗模擬驗證正確性與可行性。相關研究成果如下：

期刊論文

[1] Y. C. Liao, C. C. Lin, H. C. Chang, and C. W. Liu, "Self-compensation technique for simplified belief-propagation algorithm", *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 3061-3072, Jun. 2007

[2] Y. C. Liao, H. C. Chang, and C. W. Liu, "Carry-Estimation technique for Fixed-Width Multipliers", to be published in *Journal of VLSI Signal Processing and Systems for Signal, Image, and Video Technology*.

研討會論文

[3] Y. C. Liao, C. C. Lin, C. W. Liu and H. C. Chang, "A dynamic normalization technique for decoding LDPC codes", in *IEEE Workshop Signal Processing Systems* 2005 *(SiPS 2005)*, Nov., 2005, pp. 768-772.

其它發表論文

[4] Y. S. Wu, Y. T. Liu, H.C. Chang, Y. C. Liao, and H. C. Chang, "Early-pruned K-best sphere decoding algorithm based on radius constraints", has been accepted by 2008 *IEEE International Conference on Communications* (*ICC 2008*)

[5] H. C. Chang, Y. C. Liao, and H. C. Chang, "Low-complexity prediction techniques of K-Best sphere decoding for MIMO systems," in *IEEE Workshop on Signal Processing Systems* 2007 (*SiPS 2007*), Oct. 2007, pp. 45–49.

[6] Y. C. Liao, H. C. Chang, and C. Liu, "Carry estimation for two's complement fixed-width multipliers," in *IEEE Workshop on Signal Processing Systems Design and Implementation* 2006 (*SiPS 2006*), Oct. 2006, pp. 345 – 350.

[7] H. A. Huang, Y. C. Liao and H.C. Chang, "A self-compensation fixed-width booth multiplier and its 128-point FFT applications," in *IEEE International Symposium on Circuits and Systems* 2006 (*ISCAS 2006*), May, 2006, pp. 3538-3541.

其他尚有兩篇期刊論文在審查中。專利部分有四項美國專利與四項中華民國專利申請中。

廖彥欽同學已修滿 32 學分，總著作點數 6 點，滿足本所博士班申請畢業之相關規定第(二)項規則。此外，廖同學曾參與多項經濟部學界科專計畫，國科會，工研院與業界合作計畫。並帶領 OCEAN 研究團隊進行多項研究計畫，多次於國際重大會議發表研究成果，以團隊合作方式將研究能量發揮淋漓盡致，其研究成果備受肯定。

總言之，廖彥欽同學於博士班修業其間，已滿足本所在課業及研究上之嚴格要求，並於團隊研究方面，深獲肯定，特以推薦之。

推薦人

國立交通大學 電子工程學系 教授

張錫嘉

劉志尉

中　華　民　國　九　十　七　年　一　月

國 立 交 通 大 學

論文口試委員會審定書

本校電子工程學系電子研究所博士班 廖彥欽 君

所提論文 多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究

合於博士資格標準、業經本委員會評審認可。

口試委員：

李鎮宜	吳文榕
趙啟超	黃家齊
魏學文	吳安宇
劉志尉	張錫嘉

指導教授：張錫嘉 教授　劉志尉 教授

張錫嘉　　劉志尉

所　長：邱碧秀 教授

邱碧秀

系主任：周世傑 教授

周世傑

中華民國 97 年 2 月 1 日

Department of Electronics Engineering
& Institute of Electronics
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

Date : Feb. 1, 2008

We have carefully read the dissertation entitled

Research on Sphere\LDPC Decoder for Coded MIMO Systems

submitted by Yen-Chin Liao in partial fulfillment of the requirements of the degree of DOCTOR OF PHILOSOPHY and recommend its acceptance.

Chen-Yi Lee

Wen-Rong Wu

Chi-Chao Chao

Chia-Chi Huang

Shyue-Win Wei

An-Yeu Wu

Chih-Wei Liu

Hsie-Chia Chang

Thesis Advisor : Chih-Wei Liu

Hsie-Chia Chang

Director : Bi-Shiou Chiou

Bi-Shiou Chiou

Chairman : Shyh-Jye Jou

Shyh-Jye Jou

多輸入輸出系統之球體解碼與低密度奇偶校驗碼之研究

研究生：廖彥欽　　指導教授：張錫嘉教授、劉志尉教授

國立交通大學電子工程學系電子研究所

摘　要

本論文研究探討多輸入輸出系統之球體解碼（sphere decoding）與低密度奇偶校驗碼（low-density parity-check code, 簡稱 LDPC code）解碼方式，藉由建立機率模型之理論分析與推導，經過電腦模擬驗證，提出適用於硬體實現之高效能低複雜度演算法。

球體解碼（Sphere Decoding）可描述為一個樹狀圖上找尋最佳路徑之問題，其中 *K*-best algorithm 為一常見之簡化演算法，其固定計算量之特色適合硬體實作。但為了維持與傳統 sphere decoding 相當之效能，需要大量排序運算，造成硬體實作時複雜度大幅增加。因此我們提出了低複雜度排序法與路徑刪除（early pruning）技巧，論文中所提出之路徑刪除技巧可及早在樹狀圖上刪去對應較低機率之路徑。路徑刪除條件與相關參數，可藉由建立與通道統計特性、路徑函數（path metric）以及系統錯誤率等所對應機率模型獲得；其平均計算量亦可依循此機率模型經由理論推導得到。根據理論分析之平均計算量，我們提出 early-pruned multi-*K*-best algorithm，以進一步提升解碼速度。利用電腦模擬一

64-QAM 4×4 MIMO 系統，在維持與傳統 sphere decoding 相當之效能時，上述之方法可達到約 90%計算複雜度之改進。

信度傳播（Log belief-propagation algorithm, 簡稱 Log-BP algorithm）是常見 LDPC 碼解碼方法，其中需要之非線性運算通常以查表法或 min-sum algorithm 實現。前者需要大量硬體成本，且大量查表造成電路之時間延遲，故在設計高速解碼器多採用後者。為了減少 min-sum algorithm 由 Log-BP algorithm 簡化所造成之效能損失，我們探討並提出動態補償方法，此補償量可描述為解碼器輸入，通道雜訊，與解碼疊代次數之函數。我們進一步利用 order statistic 與 density evolution 等技巧分析並推導出此動態補償量函數，並依此提出三類可在硬體上實現之動態補償法。我們以此方法模擬 DVB-S2 系統，min-sum algorithm 加上此動態補償僅造成 5%之面積增加，最多可得到 1.0dB 之 SNR 改善。

論文最後探討 MIMO 接收端訊號偵測與通道解碼之相互影響。當系統採用如 LDPC 碼等需要以信度傳播作為解碼，sphere decoder 需要修正為 list sphere decoder，並產生一清單(candidate list)以計算通道解碼器需要之可靠度資訊。研究過程中發現，疊代解碼（iterative decoding）的收斂情形在 MIMO 環境中受到前級輸出之影響甚劇，當 candidate 過少時將造成嚴重 error floor。然而，直接利用 sphere decoding algorithm 產生較大的 candidate list 所需要之複雜度過高，因此我們提出一種增加 candidate 之方式，相形之下需要之運算較少。最後我們模擬 IEEE802.11n LDPC 碼在 64-QAM 4×4 MIMO 通道之效能，採用我們提出之清單擴增方法，搭配路徑刪除法，在降低 error floor 的同時，最多可再減少 94%之計算複雜度。

Research on Sphere/LDPC Decoder for Coded MIMO Systems

Student: Yen-Chin Liao Adviser: Hsie-Chia Chang and Chih-Wei Liu

Department of Electronics Engineering & Institute Electronics

National Chiao Tung University

Abstract

This dissertation presents algorithm designs for sphere decoders and low-density parity check (LDPC) code decoders in multi-input multi-output (MIMO) systems from implementation point of view. Based on statistical techniques, complexity reduction schemes are proposed. Sphere decoders of hard-decision outputs and LDPC decoding algorithms in AWGN channel are discussed first. Then the sphere decoders with soft-decision outputs for channel-coded MIMO systems are investigated.

Sphere decoding algorithm is one realization of maximum likelihood signal detection for MIMO systems, and the computation can vary with channel due to the fading phenomena. Among several modified algorithms, K-best algorithm is suitable for hardware implementation for the constant computation and predictable hardware complexity. However, K-best algorithm has to be realized with the assumption of worst channel condition in order to maintain the system performance. For complexity reduction, an early pruning scheme combined with K-best algorithm is presented. According to the system model and channel statistics the expected complexity can be analyzed as well. Based on the complexity analysis, an early-pruned multi-K-best algorithm is proposed by which the lowest decoding speed can be further improved. The simulation results in 64-QAM 4 × 4 MIMO channel show that 90% complexity can be reduced with imperceptible degradation in both symbol error rate and bit

error rate above 10^{-5}.

For decoding LDPC codes, min-sum algorithm is one common simplification of Log-BP algorithm, but there is a performance gap due to the approximation inaccuracy. Normalization schemes are investigated to compensate the approximation error. Moreover, the normalization factor can be described by a function of the decoder inputs, noise variance, and the decoding iteration number. The data-dependent correction terms can be analyzed and derived by order statistic and density evolution. Simulated in DVB-S2 system, the dynamic normalization schemes effectively mend the SNR loss from Log-BP algorithm to min-sum algorithm with few normalization overheads, and 1.0dB SNR improvement, which is about 95% of the SNR loss from Log-BP to min-sum algorithm, can be achieved.

For channel coded MIMO systems, a sphere decoder is modified to a list sphere decoder that generates a candidate list for computing the soft inputs. Under iterative message passing decoding, the candidate list and the soft value generation impact the decoding convergence. Sufficiently large candidate list is required to avoid error floor. Thus, a path augmentation technique is proposed by which a larger and distinct list can be employed in computing the probabilistic information of each received bit. Compared with directly generating a larger list, path augmentation performs comparatively less operations. In our simulation based on a 64-QAM 4×4 MIMO system with LDPC codes defined in IEEE802.11n, the proposed augmented-list sphere decoder based on 64-best algorithm achieves the lowest error floor and saves about 50% computations, if compared to the standard list sphere decoder which is based on 128-best algorithm. Moreover, by the proposed early pruning scheme and multi-K-best algorithm, 94% reduction in sorting complexity can be achieved.

致　　謝

回想起這四年半的博士班訓練，是一段奇妙又豐富的旅途；由於許多人的協助，我得以順利完成學業與論文。

首先感謝我的指導教授劉志尉老師，帶我進入通道編碼的領域，並給予我自由發揮的空間。同時也感謝我另一個指導教授張錫嘉老師，除了在研究上提供許多獨特的見解，老師對研究的熱情更是對我們的一種激勵。此外，感謝他所帶領的 OCEAN 團隊中的每個成員；OCEAN 就像是個溫暖的大家庭，每個成員在知識上敞開且無私的交流，讓我在團隊合作過程中學習了許多寶貴的經驗，並且給我更加開闊的想法與視野。

無論是研究或生活，特別感謝建青在這段時間的協助、關懷以及鼓勵。此外，也感謝我家人給我的關愛與支持，讓我安心無慮地完成學業。

最後，感謝所有口試委員的指教與建議。

Table of Contents

List of Tables

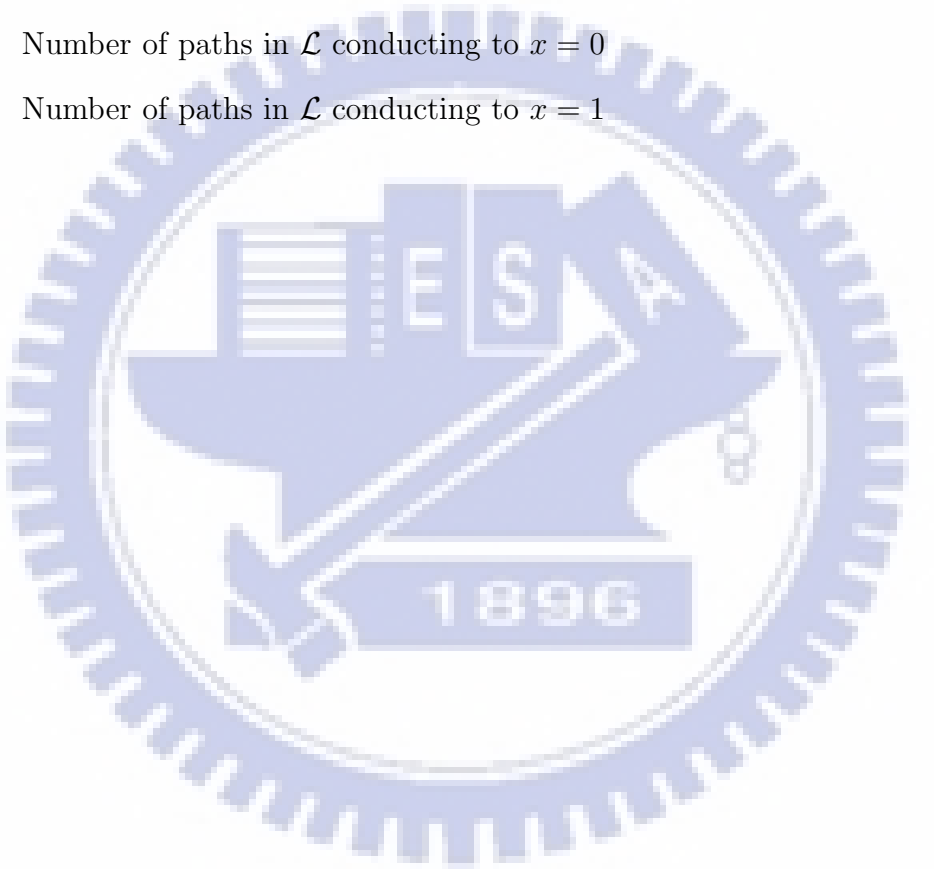
List of Figures

List of Symbols and Abbreviation

AWGN	Additive white Gaussian noise
ML	Maximum likelihood
EP-SDA	Early pruned sphere decoding algorithm
LDPC	Low-density parity-check
BP	Belief propagation
N_t	Number of transmit antennas
N_r	Number of receive antennas
M_c	Modulation level; $2M_c$ bits are mapped to 1 complex symbol
$\tilde{\mathcal{M}}(\cdot)$	Complex signal mapping function
$\mathcal{M}(\cdot)$	Equivalent real signal mapping function
$\mathbf{\Omega}$	All the constellation points of $\mathcal{M}(\cdot)$
$\mathbf{\Omega}^n$	n times Cartesian product of $\mathbf{\Omega}$
$\tilde{\mathbf{H}}$	$N_r \times N_t$ channel matrix
$\mathbf{H}$	Equivalent $2N_r \times 2N_t$ real channel matrix
$\mathbf{R}$	$2N_t \times 2N_t$ upper triangular matrix for $\mathbf{H} = \mathbf{QR}$
$\mathbf{Q}$	$2N_r \times 2N_t$ unitary matrid for for $\mathbf{H} = \mathbf{QR}$
$\tilde{\mathbf{h}}_j$	j-th column of matrix $\tilde{\mathbf{H}}$
$\tilde{\mathbf{x}}$	Transmit information vector

$\tilde{\mathbf{s}}$	$N_r \times 1$ complex transmit symbol vector
$\mathbf{s}$	Equivalent $2N_r \times 1$ real transmit symbol vector
$\tilde{\mathbf{y}}$	Complex received symbol vector
$\mathbf{y}$	Equivalent real received symbol vector
$\mathbf{q}$	$2N_r \times 1$ received symbol vector after preprocessing
$\tilde{\mathbf{v}}$	$N_r \times 1$ complex noise vector
$\mathbf{v}$	Equivalent$2N_r \times 1$ real noise vector
σ^2	noise variance
C	Radius for sphere decoding algorithm
$\mathbf{s}^{(i)}$	$(2N_t - i + 1)$-dimensional partial symbol vector of $\mathbf{s}$ for $\mathbf{s}^{(i)} = [s_i, s_{i+1}, \ldots, s_{2N_t}]$
$T(\cdot)$	Path metric
$\epsilon^{(i)}$	The i-th layer error tolerance for EP-SDA
$\boldsymbol{\Delta}$	$2N_t$-dimensional distance vector associated to the ML path
$\bar{N}^{(i)}$	The expected number of survival path at the i-th layer
α	Normalization factor in computing $\bar{N}^{(i)}$
β	Normalization factor
γ_s	Equivalent check node scaling amount in annealing compensation approach
ϵ_s	Scaling error of normalized min-sum algorithm
L	Number of buckets used in coarse-granularity sort
$\boldsymbol{H}$	Parity check matrix
BN_n	n-th bit node
CN_m	m-th check node
d_c	Check node degree
r_{mn}	The message from CN_m to BN_n

q_{nm}	The message from BN_n to CN_m
$N(m)$	The bit nodes connected to CN_m
$M(n)$	The check nodes connected to BN_n
a	Offset for compensating min-sum algorithm
$\mathcal{L}$	Candidate list
γ	Path augmentation factor; only the best γ in $\mathcal{L}$ are expended
n_0	Number of paths in $\mathcal{L}$ conducting to $x = 0$
n_1	Number of paths in $\mathcal{L}$ conducting to $x = 1$



Chapter 1

Introduction

The communication engineering is to convey information through specific channels as correctly as possible. In 1948, Claude E. Shannon proved the existence of the transmission limit, which is termed channel capacity [1,2]. It was stated that quasi-error-free transmission could be guaranteed with information rate under the channel capacity. For decades, researchers devoted much effort to approach this limit. With the advances in source coding and channel coding technology, some theoretically capacity-approaching communications have been shown achievable.

Figure 1.1 presents a general communication system block diagram where the upper and lower parts correspond to the transmitter and the receiver. Information source is first compressed by a source encoder that removes the redundancy. Subsequently specific redundant data, often referred to *parity*, is added on the compressed data for error-control. The deterministic relation between the source data and parity, algebraic structures for example, assists the receiver to detect and recover the errors occurred during transmission. All transmit medium between the transmitter and the receiver can be regarded as channels, which can be storage equipments, cables in wireline transmission, or radio links in wireless transmission. The transmitted data undergoes different corruption and interference through different channels. Thus, various modulation techniques are applied before transmitting; the data are reformed for better transmission efficiency and immunity to channel distortions. At

the receiver site, the signal detection demodulates the received signals [3]; equalization [4, 5] is sometimes required to compensate the channel effects. Provided with demodulated data or probabilistic information of received data, the channel decoder then corrects the erroneous symbols after signal detection.

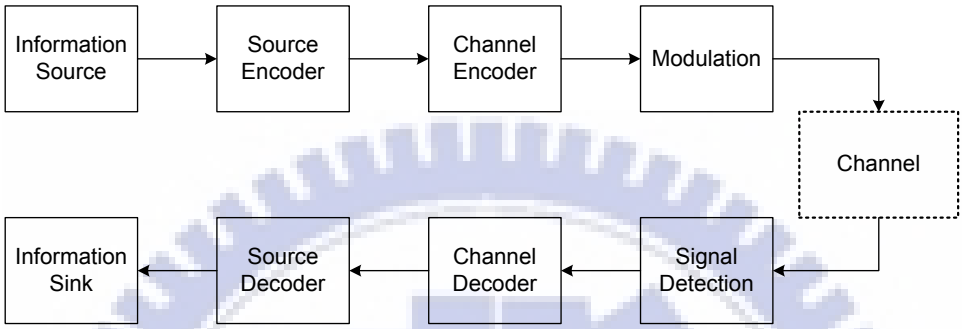


Figure 1.1: Communication system.

1.1 Channel Coding

Channel coding, or also termed error control coding, is an essential technology for reliable transmission. After the invention of turbo codes [6] and the rediscovery of low-density parity check (LDPC) codes [7–9], Shannon capacity (in additive white Gaussian noise, AWGN channel) is proved achievable by iterative decoding process [10–13]. The inherent parallelism in belief-propagation (BP) algorithm [7, 9, 14] for decoding LDPC codes facilitates high-speed LDPC decoder designs. Therefore, many advanced systems such as digital television broadcasting (DVB-S2 [15], DMB-TH [16]), wireless local area network (IEEE802.11n [17]), wireless metropolitan network (IEEE802.16e [18]), and 10G BASE-T Eithernet (IEEE802.3an [19]), all employ LDPC codes as the forward error correction (FEC) technique.

By taking logarithm of the decoder inputs, the BP algorithm is transformed to the

equivalent Log-BP decoding algorithm, and the computations can be reduced since the multiplications are transformed to additions in the logarithm domain. However, an nonlinear operation is introduced, leading to implementation difficulity. Alternatively, the min-sum algorithm [20, 21] avoids the nonlinear function but leads to performance degradation. The gap between the min-sum algorithm and the Log-BP algorithm can be reduced by a constant correction term, either normalization or offset [22–30]. Indeed, the normalization factor can be represented as a function of the decoder inputs, channel statistics, and the decoding iteration number. To further improve the error performance, we did an analysis based on the order statistic [31,32] and density evolution [33] to derive dynamic normalization factors. With little overheads in circuit implementation, we present several dynamic normalization schemes by which the normalization factors are determined on the fly.

1.2 MIMO Detection

For wireless communication, fading phenomenon [34] impacts transmission efficiency and system performance. Utilizing the fading nature of wireless channels, multi-input multi-output (MIMO) systems have emerged as powerful technologies for reliable and high-data-rate wireless transmission. The inherent diversity gain provided by the multiple channels significantly improves the signal quality and boosts the system capacity [34, 35]. However, maximum achievable diversity gain is determined by the signal detection approach [36]. Among various linear and non-linear MIMO detection schemes [34, 35, 37–43], maximum likelihood (ML) detection is shown to be capable of attaining full diversity gain. ML detection often transforms the detection to solve an integer least-squared of linear equations, which has been proved to be NP hard [44, 45].

Sphere decoding algorithm [42, 43, 46, 47] is one applicable approach to realize ML de-

tection for MIMO systems. Described by closest-point-search or tree-search problems, the sphere decoding can be classified into two major categories, depth-first search and breadth-first search. The computation of the former is channel-dependent, and the resulted non-constant decoding throughput makes hardware implementation more difficult. Due to the constant computations, the sub-optimum breadth-first search is more practicable for implementation, where parallel processing or pipelining techniques can be applied for high-throughput decoder designs. K-best algorithm [48–51] is the very representative breadth-first search realization. At each layer of the search tree, K best candidates are kept before the algorithm proceeds to the next layer. In the worst channel condition, large K is required for complicated modulation, 64-QAM for example, to maintain error performance similar to the depth-first search decoders, resulting in unmanageable sorting complexity in circuit implementation. Each of the two search strategy has its own advantages, and therefore we consider a hybrid strategy, by which a pruning scheme is applied to the breadth-first algorithms. Similar to depth-first decoders, the proposed pruning criterions are based on a set of statistically derived radii. Given the channel model and design parameters (ex. error tolerance), distinct radius constraint for each layer can be computed. Moreover, the statistical model for deriving the pruning criterions can be employed in analyzing the computation complexity of the proposed early-pruned sphere decoders.

1.3 Channel Coded MIMO System

For a channel-coded MIMO system in Figure 1.2, sphere decoding algorithm needs to be modified to list sphere decoding that generates a candidate list when probabilistic information, also termed soft values, are required as the subsequent channel decoder input. The list size is a tradeoff between error performance and computation complexity. The decoder

fails in computing the soft values when there is no sufficient candidates, and estimation for soft values is required. Path augmentation techniques [38, 49] were proposed to provide an equivalently larger list that reduces the probability of failing to compute the soft values. According to our simulation result, the candidate list size impacts the LDPC decoding convergence. Thus, we present an augmented-list sphere decoder that guarantees the augmented list always capable of delivering the soft values.

1.4 Thesis Organization

Algorithm level complexity reduction for designing sphere decoders and LDPC decoders are the focus of this dissertation. By statistical techniques, essential parameters and complexity can be analyzed, at design time, with improved simulation efficiency. The dissertation can be organized as follows. In Chapter 2, MIMO system models are introduced, and several MIMO signal detection methods are briefly reviewed. Then, the early-pruned sphere decoding algorithms are presented in Chapter 3, including parameters derivations and complexity analysis. Dynamic normalization techniques for normalized min-sum algorithm in decoding LDPC codes are presented in Chapter 4, wherein an order-statistic-based analysis combined with density evolution technique for deriving the dynamic factors is given as well. Subsequently, list sphere decoder designed for channel codes decoded by iterative algorithm is discussed; an augmented list sphere decoding with compensation is proposed. Finally, Chapter 6 concludes this work.

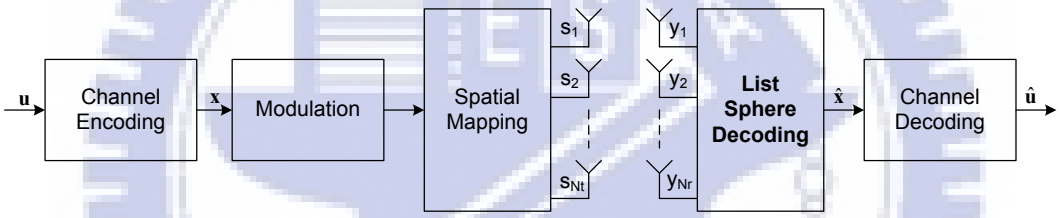


Figure 1.2: Channel Coded MIMO system.

Chapter 2

MIMO System

MIMO technology has emerged as a promising technique for reliable and high-data-rate wireless applications due to the spatial multiplexing and diversity gains. The term diversity gain refers to the slope of the error probability vesus SNR plot in a Log-Log scale. The radio links between the transmit and the receive antennas provide multiple channels and thus boost the system capacity. Thanks to the fading nature of the multiple channels, the signal replicas at the receiver can be combined, and the resulted diversity gain improves the received signal in terms of signal noise power ratio (SNR) and signal quality. Indeed, the maximum achievable diversity gain is determined by the signal detection schemes as the system spatial multiplexing strategy is given [36]. Maximum likelihood (ML) signal detection is one nonlinear, and also optimum, detection approach that fully exploits the system spatial diversity with the cost of much higher computation complexity as compared to linear schemes such as zero-forcing, minimum mean square error (MMSE) detections or successive cancellation [34, 35, 37]. In the following, a brief review of the system and the channel models will be given first, and MIMO detection schemes will be introduced later. Accordingly, in Chapter 3 and Chapter 5, the models will be applied and simulated for the study of sphere decoding algorithms, an efficient means to realize ML detection. .

2.1 System Model

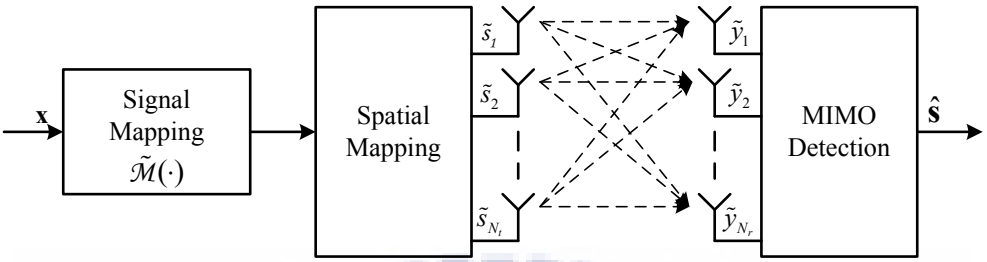


Figure 2.1: Simplified MIMO system

Figure 2.1 illustrates a simplified $N_r \times N_t$ MIMO system with N_t transmit and N_r receive antennas. The information bits $\mathbf{x}^{(t)} = [x_{1,1}^{(t)}, x_{1,2}^{(t)}, \ldots, x_{1,2M_c}^{(t)}, \ldots, x_{N_t,1}^{(t)}, \ldots, x_{N_t,2M_c}^{(t)}]^T$ are first converted to the complex signals $\tilde{\mathbf{s}}^{(t)} = [\tilde{s}_1^{(t)}, \tilde{s}_2^{(t)}, \ldots, \tilde{s}_{N_t}^{(t)}]^T$ via $\tilde{\mathcal{M}}(\cdot)$ before spatial mapping, where t is the transmit time index and $\tilde{s}_k = \tilde{\mathcal{M}}(x_{k,1}^{(t)}, x_{k,2}^{(t)}, \ldots, x_{k,2M_c}^{(t)})$. The MIMO channel is often described by matrix $\tilde{\mathbf{H}}(t,\tau)$ and

$$\tilde{\mathbf{H}}(t,\tau) = \begin{bmatrix} \tilde{h}_{1,1}(t,\tau) & \tilde{h}_{1,2}(t,\tau) & \cdots & \tilde{h}_{1,N_t}(t,\tau) \\ \tilde{h}_{2,1}(t,\tau) & \tilde{h}_{2,2}(t,\tau) & \cdots & \tilde{h}_{2,N_t}(t,\tau) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{h}_{N_r,1}(t,\tau) & \tilde{h}_{N_r,2}(t,\tau) & \cdots & \tilde{h}_{N_r,N_t}(t,\tau) \end{bmatrix}. \tag{2.1}$$

Note that τ refers to the propagation delay and $\tilde{h}_{i,j}(t,\tau)$ models the channel response between the j-th transmit and i-th receive antennas. Represented by the superposition of n_s resolvable paths between each link, the channel matrix can be further described as

$$\tilde{\mathbf{H}}(t,\tau) = \sum_{k=0}^{n_s-1} \tilde{\mathbf{H}}_{\tau_k}^{(t)} \delta(\tau - \tau_k), \tag{2.2}$$

where $\tilde{\mathbf{H}}_{\tau_k}^{(t)}$ contributes to the channel matrix of the k-th delay path. Thus the received signals $\tilde{\mathbf{y}}(t,\tau)$ can be represented by

$$\tilde{\mathbf{y}}(t,\tau) = \begin{bmatrix} \tilde{y}_1(t,\tau) \\ \tilde{y}_2(t,\tau) \\ \cdots \\ \tilde{y}_1(t,\tau) \end{bmatrix} = \sum_{k=0}^{n_s-1} \tilde{\mathbf{H}}_{\tau_k}^{(t)} \tilde{\mathbf{s}}^{(t)} \delta(\tau - \tau_k). \tag{2.3}$$

Each element in the matrix $\tilde{\mathbf{H}}_{\tau_k}^{(t)}$ is usually determined by several factors in physical propagation such as antenna patterns, antenna spacing, and directions of signal arrival(or departure), etc. More channel models are detailed in [34, 52], and *uncorrelated flat fading* is one common and simple MIMO channel model among them. The channel response between the j-th transmit antenna and the i-th receive antenna is modeled as a single-path narrow band Rayleigh fading channel. That is, $n_s = 1$ in (2.2) and each $h_{i,j}(t,\tau)$ can be modeled by a circular Gaussian random variable $\mathcal{CN}(0,1)$ [34, 53]. As a result, the channel model is irrelevant to the delay τ, and (2.1) is reduced to

$$\tilde{\mathbf{H}}(t,\tau) = \tilde{\mathbf{H}}^{(t)} = \begin{bmatrix} \tilde{h}_{1,1}^{(t)} & \tilde{h}_{1,2}^{(t)} & \cdots & \tilde{h}_{1,N_t}^{(t)} \\ \tilde{h}_{2,1}^{(t)} & \tilde{h}_{2,2}^{(t)} & \cdots & \tilde{h}_{2,N_t}^{(t)} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{h}_{N_r,1}^{(t)} & \tilde{h}_{N_r,2}^{(t)} & \cdots & \tilde{h}_{N_r,N_t}^{(t)} \end{bmatrix} \tag{2.4}$$

where all $\tilde{h}_{i,j}^{(t)}$ are independent, identically distributed (i.i.d.) circular Gaussian random variables. Accordingly, the relation between the transmit and receive signals becomes

$$\tilde{\mathbf{y}}^{(t)} = \tilde{\mathbf{H}}^{(t)} \tilde{\mathbf{s}}^{(t)}. \tag{2.5}$$

Note that (2.5) only considers the impacts on signal propagation, a more general model should be

$$\tilde{\mathbf{y}}^{(t)} = \tilde{\mathbf{H}}^{(t)}\tilde{\mathbf{s}}^{(t)} + \tilde{\mathbf{v}}^{(t)}, \tag{2.6}$$

where $\tilde{\mathbf{v}}^{(t)} = [\tilde{v}_1^{(t)}, \tilde{v}_2^{(t)}, \ldots, \tilde{v}_{N_r}^{(t)}]^T$ is the receiver additive noise vector and the $\tilde{v}_k^{(t)}$'s are i.i.d. circular Gaussian random variables $\mathcal{CN}(0, \sigma_v^2)$.

Equation (2.6) applies to many linear space-time codes. Besides, $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{v}}$ can be further replaced by an $N_r \times T_c$ matrix, and $\tilde{\mathbf{s}}$ by an $N_t \times T_c$ matrix, to denote block transmission for T_c time interval. The system model (2.6) is still applicable as long as the channel remains unchanged during the T_c time slots. For simplicity, the symbol time index t will be omitted henceforth. The spatial mapping will be referred to pure spatial multiplexing by which the complex signal $\tilde{s}_k$ will be directed to the k-th transmit antenna. Furthermore, the channel matrix $\tilde{\mathbf{H}}$ is assumed to have full rank and to be perfectly estimated at the receiver.

2.2 MIMO Signal Detection Algorithms

MIMO signal detection can be classified into linear detection and nonlinear detection [34,37], and both approaches are often reduced to finding the integer least-squared solution for N_r sets of N_t-dimensional linear equations. Linear equalization and successive interference cancellation are two representative approaches in the linear category, by which an unconstraint least-squared solution is found and then quantized to the nearest integer values. For non-linear detection, maximum-likelihood detection can achieve optimum performance with the expense of higher computation complexity. In addition, iterative detection and channel decoding should be another category [38–41]. Either linear or non-linear detection can be applied to provide the probabilistic information for iterative process between the MIMO

detector and the channel decoder.

2.2.1 Linear Equalization

In an SISO system, impaired received signal can be compensated by equalizing the channel response. Zero-forcing and minimum mean-squared error (MMSE) equalizations are the two most common linear schemes. The same concept can also be applied to MIMO signal detection; the transmitted $\tilde{\mathbf{s}}$ can be recovered from $\tilde{\mathbf{y}}$ by directly equalizing the channel effects.

By *singular value decomposition* [54], the channel matrix $\tilde{\mathbf{H}}$ can be factorized into

$$\tilde{\mathbf{H}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \tag{2.7}$$

where $\mathbf{\Sigma}$ is an $N_r \times N_t$ matrix with elements $\sigma_{j,j} = \sqrt{\lambda_j}$ for $j = 1, 2, \ldots, N_r$, and λ_j is the j-th eiganvalue of $\mathbf{H}^H\mathbf{H}$. $\mathbf{U}$ and $\mathbf{V}$ have dimensions $N_r \times N_r$ and $N_t \times N_t$ respectively; the columns of $\mathbf{U}$ are the eiganvetors of $\mathbf{H}^H\mathbf{H}$ and the columns of $\mathbf{V}$ are eiganvectors of $\mathbf{H}\mathbf{H}^H$. Note that $\mathbf{U}^H\mathbf{U} = \mathbf{I}_{N_r}$ and $\mathbf{V}\mathbf{V}^H = \mathbf{I}_{N_t}$. The pseudo-inverse channel matrix $\tilde{\mathbf{H}}^+ = (\tilde{\mathbf{H}}^H\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^H$ can be derived by

$$\tilde{\mathbf{H}}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^H, \tag{2.8}$$

where $\mathbf{\Sigma}^+$ can be computed by transposing $\mathbf{\Sigma}$ then replacing the diagonal with $\frac{1}{\sigma_{j,j}}$. Moreover, $\mathbf{\Sigma}^+\mathbf{\Sigma} = \mathbf{I}_{N_t}$; when $N_r = N_t$, $\tilde{\mathbf{H}}^+ = \tilde{\mathbf{H}}^{-1}$, i.e. the inverse of the channel matrix.

Zero-forcing (ZF) equalization can be realized by directly multiply the received vector $\tilde{\mathbf{y}}$

by $\tilde{\mathbf{H}}^{+}$; as a result,

$$\begin{aligned}\tilde{\mathbf{H}}^{+}\tilde{\mathbf{y}} &= (\mathbf{V}\boldsymbol{\Sigma}^{+}\mathbf{U}^{H})(\tilde{\mathbf{H}})\tilde{\mathbf{s}} + (\mathbf{V}\boldsymbol{\Sigma}^{+}\mathbf{U}^{H})\tilde{\mathbf{v}} \\ &= (\mathbf{V}\boldsymbol{\Sigma}^{+}\mathbf{U}^{H})(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{H})\tilde{\mathbf{s}} + (\mathbf{V}\boldsymbol{\Sigma}^{+}\mathbf{U}^{H})\tilde{\mathbf{v}} \\ &= \tilde{\mathbf{s}} + \tilde{\mathbf{H}}^{+}\tilde{\mathbf{v}}. \end{aligned} \tag{2.9}$$

The ZF solution can be derived by quantizing $\tilde{\mathbf{H}}^{+}\tilde{\mathbf{y}}$ to its nearest integers. As shown in (2.9), the noise is scaled by $\tilde{\mathbf{H}}^{+}$. The effective noise power can be computed by

$$\begin{aligned}E[(\tilde{\mathbf{H}}^{+}\tilde{\mathbf{v}})^{H}(\tilde{\mathbf{H}}^{+}\tilde{\mathbf{v}})] &= (\tilde{\mathbf{H}}\tilde{\mathbf{H}}^{H})^{-1}E[\tilde{\mathbf{v}}^{H}\tilde{\mathbf{v}}] \\ &= \sigma_v^2(\tilde{\mathbf{H}}^{H}\tilde{\mathbf{H}})^{-1} \end{aligned} \tag{2.10}$$

The potentially reduced SNR and degradation from the resulted noise enhancement limits the system performance. Moreover, it can be shown that the maximum achievable diversity gain is $N_r - N_t + 1$ [36], provided that $N_r \geq N_t$ and very high probability of $\tilde{\mathbf{H}}$ having full rank.

MMSE equalization aims to substitute the $\tilde{\mathbf{H}}^{+}$ in (2.9) by other compensation matrix such that the average enhanced noise power is minimized, which is equivalent to maximizing the detector output SNR. Given ρ as the received SNR, the MMSE equalization estimates $\tilde{\mathbf{s}}$ by multiplying $\tilde{\mathbf{y}}$ with

$$\mathbf{D}_{MSSE} = \left(\frac{\mathbf{I}_{N_r}}{\rho} + \tilde{\mathbf{H}}^{H}\tilde{\mathbf{H}}\right)^{-1}\tilde{\mathbf{H}}^{H}. \tag{2.11}$$

When the receive SNR ρ is high, the $\mathbf{D}_{MSSE}$ (2.11) approaches to $(\tilde{\mathbf{H}}^{H}\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^{H} = \tilde{\mathbf{H}}^{+}$. That is, the MMSE detection reduces to zero forcing at high SNR region. MMSE detection improves the error performance at low SNR region, but has the same diversity gain of

the zero-forcing equalization, which is at most $N_r - N_t + 1$. Moreover, MMSE equalizer requires accurate received SNR estimation for deriving ρ in (2.11) and computation of matrix inversion, leading to higher hardware complexity.

2.2.2 Successive Interference Cancelling

The MIMO system described in (2.6) can be rewritten as

$$\begin{aligned} \tilde{\mathbf{y}} &= \sum_{j=1}^{N_t} \tilde{\mathbf{h}}_j \tilde{s}_j + \tilde{\mathbf{v}} \\ &= \tilde{\mathbf{h}}_k \tilde{s}_k + \sum_{j=1, j\neq k}^{N_t} \tilde{\mathbf{h}}_j \tilde{s}_j + \tilde{\mathbf{v}}, \end{aligned} \tag{2.12}$$

where $\tilde{\mathbf{h}}_j$ denotes the j-th column of the channel matrix $\tilde{\mathbf{H}}$. The second term of (2.12) can be regarded as interference to $\tilde{s}_k$. Subtracting the partially detected symbols from $\tilde{\mathbf{y}}$ makes it easier to detect the rest undetected symbols, provided that the probability of correctly estimating these partial symbols is very high. Similar to *decision-feedback equalization* in an SISO system, probability of correctly estimating the rest undetected symbols increases since some of the interference are removed. Besides, the computation complexity of jointly decoding the whole vector could be much higher than that of estimating partial symbols. By this divide-and-conquer strategy, successive interference cancelling (SIC) reduces the computation of decoding one high-dimensional vector to several less complicated operations. SIC could suffer from severe error propagation if the first few symbols are detected incorrectly. Thus, proper ordering is required for SIC to achieve better error performance [55, 56]. The symbols with larger signal strength should be detected earlier. After ordering, SIC completes the detecting in N_t stages. At the k-th stage, the $\tilde{s}_k$ can be detected after the following two

steps:

- **Interference cancellation**: Let $\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{k-1}$ be the estimates of $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{k-1}$. By subtracting them from $\tilde{\mathbf{y}}$, the less interfered received vector $\tilde{\mathbf{y}}^{(k)}$ will be

$$\tilde{\mathbf{y}}^{(k)} = \tilde{\mathbf{y}} - \sum_{j=1}^{k-1} \hat{s}_j \tilde{\mathbf{h}}_j + \tilde{\mathbf{v}}. \tag{2.13}$$

  Moreover, it can be verified that

$$\tilde{\mathbf{y}}^{(k)} = \tilde{\mathbf{y}}^{(k-1)} - \hat{s}_{k-1} \tilde{\mathbf{h}}_{k-1}. \tag{2.14}$$

- **Interference nulling**: After the previous step, the interference from $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{k-1}$ is removed. Interference nulling will suppress the interference from $\tilde{s}_{k+1}, \tilde{s}_{k+2}, \ldots, \tilde{s}_{N_t}$ to derive $\hat{s}_k$. The nulling process is equivalent to solving N_r sets of $(N_t - k + 1)$-dimensional linear equations, and thus, the aforementioned zero-forcing or MMSE equalization approaches can be applied. Since only one symbol is decoded at this stage, the nulling process only requires the k-th row of the zero-forcing pseudo-inverse matrix, denoted by $\mathbf{Z}_k^+$, or the MMSE matrix, which is $\left(\frac{\mathbf{I}_{N_r}}{\rho} + \mathbf{Z}_k^H \mathbf{Z}_k\right)^{-1} \mathbf{Z}_k^H$, for computing $\hat{s}_k$. This row vector will be referred as nulling vector. Then $\hat{s}_k$ can be obtained by computing the inner product of $\tilde{\mathbf{y}}^{(k)}$ and the nulling vector. Note that $\mathbf{Z}_k$ is derived by replacing the first $k-1$ rows of $\tilde{\mathbf{H}}$ by zero, and $\mathbf{Z}_k^+$ is the pseudo-inverse matrix of $\mathbf{Z}_k$.

  Another common approach to suppress the interference is to subtract the projection of $\tilde{\mathbf{y}}^{(k)}$ on $\mathbf{b}_{k+1}, \mathbf{b}_{k+2}, \ldots, \mathbf{b}_{N_t}$, where $\mathbf{b}_{k+1}, \mathbf{b}_{k+2}, \ldots, \mathbf{b}_{N_t}$ are the orthonormal basis of the subspaces created by $\tilde{\mathbf{h}}_{k+1}, \tilde{\mathbf{h}}_{k+2}, \ldots, \tilde{\mathbf{h}}_{N_t}$. The orthonormal basis can be derived via *Gram-Schmidt orthonormalization procedure* [54]. Accordingly, the resulted vector

$\hat{\mathbf{y}}^{(k)}$ for deriving $\hat{s}_k$ can be obtained through

$$\hat{\mathbf{y}}^{(k)} = \tilde{\mathbf{y}}^{(k)} - \sum_{j=k+1}^{N_r} < \tilde{\mathbf{y}}^{(k)}, \mathbf{b}_j > \mathbf{b}_j, \tag{2.15}$$

where $< \mathbf{a}_1, \mathbf{a}_2 >$ denotes inner products of vectors $\mathbf{a}_1$ and $\mathbf{a}_2$. Subsequently, detecting $\tilde{s}_k$ from $\hat{\mathbf{y}}^{(k)}$ becomes a SIMO detection problem, maximum ratio combining or equal-gain combining schemes can be applied to obtain $\hat{s}_k$ [34, 37].

For $k > 1$, the effective channel matrix that an SIC detector deals with has smaller dimension than $\tilde{\mathbf{H}}$ does, the enhanced noise power is smaller, leading to better error performance. Bell Lab layerd space-time (BLAST) architectures [35, 57–59] can be categorized in this type. Moreover, the diversity gain has been proved to be greater than $N_r - N_t + 1$. In fact, the maximum achievable diversity gain varies with k, which is $N_r - N_t + k$.

2.2.3 Maximum-likelihood Signal Detection

Based on the system model $\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \tilde{\mathbf{v}}$ described in (2.6), maximum-likelihood (ML) signal detection estimates the transmit vector $\tilde{\mathbf{s}}$ by searching for a vector $\hat{\mathbf{s}}$ that maximizes the conditional probability

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s}' \in \tilde{\mathbf{\Omega}}^{N_t}} Pr(\tilde{\mathbf{y}} | \mathbf{s}'), \tag{2.16}$$

where $\tilde{\mathbf{\Omega}}$ denotes all possible constellation points of the mapping function $\tilde{\mathcal{M}}(\cdot)$. Following the Gaussian noise in channel model (2.6), (2.16) can be further reduced to a closest-lattice-point searching problem [47],

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}' \in \tilde{\mathbf{\Omega}}^{N_t}} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\mathbf{s}'\|^2, \tag{2.17}$$

where each $\mathbf{s}'$ denotes an N_t-dimensional lattice point of the lattice generated by $\tilde{\mathbf{H}}$. Note that $\|\mathbf{a}\|^2 = \sum_{k=1}^{N} a_k^2$ represents Euclidean-norm of the N-dimensional vector $\mathbf{a}$, and $\tilde{\mathbf{\Omega}}^{N_t} = \tilde{\mathbf{\Omega}} \times \tilde{\mathbf{\Omega}} \times \cdots \times \tilde{\mathbf{\Omega}}$, the N_t times Cartesian product of $\tilde{\mathbf{\Omega}}$. ML detection has been proved to be one MIMO detection scheme that fully utilizes the benefit of diversity, and has been applied to analyze performance of many systems and space-time codes. However, it is perceived in (2.17) that the computation complexity increases exponentially with $N_t \times |\tilde{\mathbf{\Omega}}|$. Thus, efficient searching algorithms to realize ML detection are desired for high performance MIMO systems. Sphere decoding algorithm [42, 43] and sequential detection [38] are two applicable approaches. However, the efficiency of these algorithms depends on the searching strategy, which is still challenging in hardware implementation.

Chapter 3

Sphere Decoding Algorithm

For all the schemes introduced in Chapter 2, the computation complexity are ranked in an ascending order by ZF, MMSE, ZF-SIC, MMSE-SIC, ML, and the order is the same for the error performance and the achievable diversity gain. Realization of ML detection for a high performance system is still very challenging, exhaustively searching for the minimizer in (2.17) or the maximizer in (2.16) is infeasible. In fact, ML detection often transforms the detection to solving an integer least-squared of linear equations, which has been proved to be NP hard [44, 45]. Alternatively, sphere decoding algorithm was proposed and proved to have tractable polynomial complexity [42, 43, 46, 47, 60, 61]. Thus real-tim ML detection is still applicable. However, the complexity depends on the efficiency of the search strategy. Several variation of sphere decoding algorithm will be introduced in the following.

3.1 Sphere Decoding Algorithm

The complex system model in (2.6) is often rearranged into the real-valued form by

$$\begin{aligned}\mathbf{y} &= \begin{bmatrix} \Re\{\tilde{\mathbf{y}}\} \\ \Im\{\tilde{\mathbf{y}}\} \end{bmatrix} \\ &= \begin{bmatrix} \Re\{\tilde{\mathbf{H}}\} & -\Im\{\tilde{\mathbf{H}}\} \\ \Im\{\tilde{\mathbf{H}}\} & \Re\{\tilde{\mathbf{H}}\} \end{bmatrix} \begin{bmatrix} \Re\{\tilde{\mathbf{s}}\} \\ \Im\{\tilde{\mathbf{s}}\} \end{bmatrix} + \begin{bmatrix} \Re\{\tilde{\mathbf{v}}\} \\ \Im\{\tilde{\mathbf{v}}\} \end{bmatrix} \\ &= \mathbf{H}\mathbf{s} + \mathbf{v}, \end{aligned} \tag{3.1}$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ respectively refer to the real and the imaginary parts of a complex signal. The complex modulation $\tilde{\mathcal{M}}(\cdot)$ also is decomposed into two real-valued signal mapping $\mathcal{M}(\cdot)$. For instance, M^2-QAM mapping is transformed two M-PAM modulation. Then (2.17) becomes

$$\hat{\mathbf{s}}_{ML} = \arg \min_{\mathbf{s}' \in \mathbf{\Omega}^{2N_t}} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2. \tag{3.2}$$

A sphere decoder searches for the minimizer in the hypersphere $\|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 < C$, and the ML solution can be obtained by

$$\hat{\mathbf{s}}_{ML} \approx \hat{\mathbf{s}}_{SD} = \arg \min_{\mathbf{s}' \in \mathbf{\Omega}^{2N_t}, \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 \le C} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2, \tag{3.3}$$

provided that the radius is properly selected such that the sphere contains at least one lattice point.

Preprocessing on $\mathbf{y}$ can further transform the problem into a tree-search problem. With QR-decomposition [54], for instance, the channel matrix is decomposed to $\mathbf{H} = \mathbf{Q}\mathbf{R}$. Mul-

tiplying $\mathbf{y}$ by $\mathbf{Q}^T$, we can transform (3.3) to

$$\hat{\mathbf{s}}_{SD} = \arg \min_{\mathbf{s}' \in \mathbf{\Omega}^{2N_t}, \|\mathbf{q}-\mathbf{R}\mathbf{s}'\|^2 \leq C} \|\mathbf{q} - \mathbf{R}\mathbf{s}'\|^2, \tag{3.4}$$

where $\mathbf{q} = [q_1, q_2, \ldots, q_{2N_t}] = \mathbf{Q^T y}$.

The *path metric* defined on Euclidean-norm of each lattice point $\mathbf{s}'$ can be calculated by

$$\|\mathbf{q} - \mathbf{R}\mathbf{s}'\|^2 = \sum_{i=1}^{2N_t} \left(q_i - \sum_{j=i}^{2N_t} R_{i,j} s_j^{(i)} \right)^2 \tag{3.5}$$

$$= \sum_{i=1}^{2N_t} e(\mathbf{s}^{(i)}), \tag{3.6}$$

where the *partial path* $\mathbf{s}^{(i)}$ is a subset of $\mathbf{s}'$ and $\mathbf{s}^{(i)} = [s_i, s_{i+1}, \ldots, s_{2N_t}]$. Moreover, the *partial Euclidean distance* (PED) of $\mathbf{s}^{(i)}$, $T(\mathbf{s}^{(i)})$, is defined by

$$\begin{aligned} T(\mathbf{s}^{(i)}) &= \sum_{i'=i}^{2N_t} \left(q_{i'} - \sum_{j=i'}^{2N_t} R_{i',j} s_j \right)^2 \\ &= \sum_{i'=i+1}^{2N_t} \left(q_{i'} - \sum_{j=i'}^{2N_t} R_{i',j} s_j \right)^2 + \left(q_i - \sum_{j=i}^{2N_t} R_{i,j} s_j \right)^2 \\ &= T(\mathbf{s}^{(i+1)}) + e(\mathbf{s}^{(i)}). \end{aligned} \tag{3.7}$$

Accordingly, the search algorithm starts from the $2N_t$-th layer of the tree, which will be termed *root node*, to the 1-st layer of the tree, which will be termed *leaf node*. Each element of every $\mathbf{s}'$ refers to a distinct node of the tree. The number of nodes visited during the searching procedure determines the computation complexity.

3.1.1 Depth-First Search and Breadth-First Search

From (3.4) and (3.5), the lattice points in the hypersphere should satisfy the following constraint

$$\begin{aligned} C &\geq (q_{2N_t} - Rr_{2N_t,2N_t}s_{2N_t})^2 \\ &+ (q_{2N_t-1} - R_{2N_t-1,2N_t-1}s_{2N_t-1} - R_{2N_t-1,2N_t}s_{2N_t})^2 \\ &\vdots \\ &+ (q_1 - R_{1,1}s_1 - R_{1,2}s_2 - \cdots - R_{1,2N_t}s_{2N_t})^2. \end{aligned} \tag{3.8}$$

Therefore, $C \geq (q_{2N_t} - R_{2N_t,2N_t}s_{2N_t})^2$ and s_{2N_t} is confined in the range

$$\left\lceil \frac{-C + q_{2N_t}}{R_{2N_t,2N_t}} \right\rceil \leq s_{2N_t} \leq \left\lfloor \frac{C + q_{2N_t}}{R_{2N_t,2N_t}} \right\rfloor \tag{3.9}$$

by the lower bound

$$LB_{2Nt}(s_{2N_t}) = \left\lceil \frac{-C + q_{2N_t}}{R_{2N_t,2N_t}} \right\rceil \tag{3.10}$$

and the upper bound

$$UB_{2Nt}(s_{2N_t}) = \left\lfloor \frac{C + q_{2N_t}}{R_{2N_t,2N_t}} \right\rfloor. \tag{3.11}$$

Subsequently, for any s_{2N_t} of this range, the range for s_{2N_t-1} will be derived similarly, and so are the ranges of the nodes extended from them. That is,

$$\begin{aligned} LB_k(s_k|\mathbf{s}^{(k+1)}) &= \left\lceil \frac{-C_k(\mathbf{s}^{(k+1)}) + q_{k|k+1}}{R_{k,k}} \right\rceil \leq s_k \\ &\leq \left\lfloor \frac{C_k(\mathbf{s}^{(k+1)}) + q_{k|k+1}}{R_{k,k}} \right\rfloor = UB_k(s_k|\mathbf{s}^{(k+1)}) \end{aligned} \tag{3.12}$$

where $s_k|\mathbf{s}^{(k+1)}$ denotes the node on the k-th layer extended from the partial path $\mathbf{s}^{(k+1)}$, $q_{k|k+1}$ is defined by

$$q_{k|k+1} \triangleq q_k - \sum_{i=k+1}^{2N_t} R_{i,i} s_i^{(k+1)}, \tag{3.13}$$

and $C_k(\mathbf{s}^{(k+1)})$ is the *partial radius* for $\mathbf{s}^{(k+1)}$, i.e.,

$$C_k(\mathbf{s}^{(k+1)}) = C - T(\mathbf{s}^{(k+1)}). \tag{3.14}$$

According to the notations introduced above, the original Fincke and Pohst [42] searching algorithm can be described as

- **Input**: $\mathbf{q}, \mathbf{R}, C, \Omega = \{\omega_1, \omega_2, \ldots, \omega_M\}$, where $\omega_l < \omega_{l+1}$ and $\Omega^{-1}(\omega_l) = l$ for all $l = 1, 2, \ldots, M-1$.
- **Step0** (Initialization): $k = 2N_t, d_k^2 = C, q_{k|k+1} = q_{2N_t}$.
- **Step1** (Computing the range): $LB_k = \left\lceil \frac{-C_k + q_{k|k+1}}{R_{k,k}} \right\rceil$, $UB_k = \left\lfloor \frac{C_k + q_{k|k+1}}{R_{k,k}} \right\rfloor$, and $l_k = \Omega^{-1}(LB_k) - 1, s_k = \omega_l$.
- **Step2** (Radius check): $l_k = l_k + 1, s_k = \omega_{l_k}$. If $s_k \leq UB_k$, go to **Step 4**; else, go to **Step3**.
- **Step3** (Move to upper layer): $k = k + 1$. If $k = 2N_t + 1$, terminate algorithm; else, go to **Step2**.
- **Step4** (Move to lower layer): $k = k - 1$. If $k = 0$, go to **Step5**; else, $q_{k|k+1} = q_k - \sum_{j=k+1}^{2N_t} R_{k,j} s_j^{(k+1)}$, $C_k = C - T(\mathbf{s}^{(k+1)})$, then go to **Step1**.
- **Step5** (One candidate found): Record the $\mathbf{s}$ and its corresponding $T(\mathbf{s})$. Then Go to **Step2**.

As we can observe in the above algorithm, the algorithm starts from a root node, and the search moves upward if the the current path metric exceeds the upper bound UB_k; otherwise, the search proceeds downward. The search direction goes back and forth, and therefore the algorithm is also referred to as *depth-first search*.

Efficient hardware implementation of depth-first sphere decoding algorithm becomes difficult since the computation highly depends on the channel, and the non-constant computation restricts the decoder throughput. Moreover, the two-way searching direction makes it more challenging to apply parallel computing or pipelining techniques [62] to improve decoder throughput. Consequently, K-best algorithm [48, 49] similar to the M-algorithm in sequential decoding [63] was proposed. K-best algorithm modified the original algorithm by its search direction. The K-best algorithm starts from the root-layer, and only the nodes corresponding to the K smallest PEDs are kept before the algorithm proceeds to the subsequent lower layer. When the search moves to the subsequent lower layer, each one of the retained K best partial paths (parent nodes) is expanded to M_c paths (child nodes), and totally $M_c \times K$ partial paths' PEDs will be computed and compared for the new K best PEDs. The same operations continues until the first layer is reach. Hence, the search in the algorithm becomes uni-direction, which is referred to as *breadth-first search*. Because the modified algorithm only searches for local minimas at each layer, the K-best decoder may not always returns the true minimizer in (3.3), leading to performance degradation when K is too small. However, the constant computation at each layer and the recursively derived path metric described in (3.6) and (3.7) make K-best algorithm more suitable for VLSI implementation [49–51].

3.1.2 Complexity Reduction Techniques

It is perceived that the computation complexity for a depth-first sphere decoding depends on the channel, i.e. $\mathbf{R}$ and the noise variance, as well as the radius C selection of the hypersphere. The computation complexity of a bread-first sphere decoder is dominated by the value K and the sorting operations for keeping the K best PEDs.

For depth-first strategy, if the chosen radius C is too large, too many nodes will be examined, leading to much redundant computation. But if C is chosen too small, chances are all the nodes will be pruned during the search process. In this case, C should be modified to a larger value and the algorithm will start over. Therefore, the computation complexity is highly related to the selection of the radius. One straightforward choice of the initial radius is the Euclidean norm corresponding to the *Babai point* [47], which is denoted by $\hat{\mathbf{s}}_B = [\hat{s}_{B_1}, \hat{q}_{B_2}, \ldots, \hat{q}_{B_{2N_t}},]^T$ and can be derived by

$$\begin{aligned}
\hat{q}_{B_{2N_t}} &= \Omega_q^{-1}\left(\frac{q_{2N_t}}{R_{2N_t,2N_t}}, \Omega\right) \\
\hat{q}_{B_{2N_t-1}} &= \Omega_q^{-1}\left(\frac{q_{2N_t-1} - R_{2N_t-1,2N_t}\hat{s}_{B_{2N_t}}}{R_{2N_t-1,2N_t-1}}, \Omega\right) \\
&\cdots \\
\hat{q}_{B_1} &= \Omega_q^{-1}\left(\frac{q_1 - \sum_{j=2}^{2N_t} R_{1,j}\hat{s}_{B_j}}{R_{1,1}}, \Omega\right),
\end{aligned} \tag{3.15}$$

where the function $\Omega_q^{-1}(x, \Omega)$ returns the constellation point in Ω which is nearest to x. Thus, the radius C should be

$$C = \|\mathbf{q} - \mathbf{R}\hat{\mathbf{s}}_B\|^2. \tag{3.16}$$

In fact, the radius can vary throughout decoding. When a leaf node is reached, its path metric never exceeds C. Therefore, we can always update the radius to the current path

metric whenever a new lattice point satisfying the sphere constraint is found. As a result, the radius shrinks each time the searching proceeds to a leaf node, allowing more paths to be pruned. With this radius updating concept, Schnorr and Euchnorr made a small but significant modification to the original Fincke-Pohst search strategy [46]. Unlike the Fincke-Pohst strategy checking the nodes on the k-th layer, within the range LB_k and UB_k, with the order

$$\omega_l, \omega_{l+1}, \ldots,$$

the Schnorr-Euchnorr approach checks the nodes with an ascending order of

$$\hat{s}_k, \hat{s}_k - 1, \hat{s}_k + 1, \hat{s}_k - 2, \hat{s}_k + 2, \ldots,$$

with

$$\hat{s}_k = \Omega_q^{-1}\left(\frac{q_k - \sum_{j=k+1}^{2N_t} R_{k,j}\hat{s}_j}{R_{k,k}}, \Omega\right). \tag{3.17}$$

That is, the nodes corresponding to smaller PEDs will be examined earlier, and the search is quaranteed to reache the leaf nodes more quickly. Moreover, the radius shrinks in a faster rate, resulting in more early-pruned nodes. Other radius shrinking techniques to accelerate the algorithm convergence rate can be further referred to [64–66],

For bread-first search such as K-best algorithm, computation remains constant if K is constant throughout decoding. The sorting operation directly relates to the complexity, and choosing smaller K is a straightforward approach to reduce complexity; however, the error performance may degrade. Since the K-best algorithm only searches for local minimums at each layer, the probability of the ML-path being discarded increases when the channel in low SNR conditions. It was pointed out in [67] that an adaptive K can effectively reduce the computation complexity. With a signal quality indicator that is defined by the ratio of

the second minimum and the minimum of the PEDs, a larger K is employed when the ratio exceeds some threshold; otherwise, a smaller K-value is applied.

Statistic pruning, similar to the T-algorithm [63] in sequential decoding, is another type of complexity reduction technique and can be applied to both depth-first and breadth-first searching strategies. The paths with PEDs exceeding some thresholds will be ignored. More details about the pruning schemes can be referred to [68–72].

3.2 Early-Pruned Breadth-First Sphere Decoding Algorithm

Constant throughput and predictable complexity is the major advantages of breadth-first sphere decoding algorithms, however, the decoder is often designed based on the worst channel assumption to avoid performance degradation. In K-best algorithm, to achieve a high probability of finding the minimizer in (3.3) by searching for the local minima, the value K is usually large for complicated (dense) constellations, 64-QAM for instance. When the received signals are severely impaired, which results in many small PEDs, large K can prevent dropping the ML path, and therefore the average computation of K-best algorithm is usually higher.

Pruning less likely pathes is one effective approach for complexity reduction. A depth-first decoder inherently performs tree-pruning by its radius constraint. We can employ the similar technique to a breadth-first decoder by setting an upper bound at each layer for the PEDs. Although the computation is no longer constant, the single-direction data flow of the breadth-first nature still corresponds to manageable complexity. With K-best algorithm, the computation complexity remains predictable.

In the following, a K-best sphere decoder with radius constraints will be presented, including the derivation of the radius for each layer. Based on the statistical model, an early-pruned multi-K-best sphere decoder, where distinct K's are assigned to each decoding layer, is presented to improve the decoding efficiency. Since the radii equivalently exhibit the data dynamic range of the PEDs, a coarse-granularity sorting strategy can be applied for further complexity reduction.

3.2.1 Pruning Criterion

Let $n = 2N_t$ be the dimension of $\mathbf{s}$, we wish to find a set of radii $C^{(n)}, C^{(n-1)}, \ldots, C^{(1)}$ for a breadth-first decoder such that the i-th layer nodes are pruned when their corresponding PEDs exceed $C^{(i)}$. The radius $C^{(i)}$ is derived according to the error tolerance $\epsilon^{(i)}$ for

$$\Pr(T_{ML}^{(i)} > C^{(i)}) \leq \epsilon^{(i)}, \tag{3.18}$$

where $T_{ML}^{(i)} = T(\mathbf{s}_{ML}^{(i)})$ is the PED corresponds to the ML path defined in (3.7). Thus, when the distribution of the ML path is known, the radii $C^{(i)}$ can be derived under the error tolerance $\epsilon^{(i)}$ for $i = 1, 2, \ldots, n$.

Corollary 3.1. If $\mathbf{v}$ is an i.i.d. Gaussian vector of dimension n and $v_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, 2, \ldots, n$, $\mathbf{Q}$ is an $n \times m$ unitary matrix, then $\mathbf{r} = \mathbf{Q}^T\mathbf{v}$ is also an i.i.d. Gaussian vector with $r_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, 2, \ldots, n$.

Corollary 3.2. If $\mathbf{v}$ is an i.i.d. Gaussian vector of dimension d and $v_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, 2, \ldots, d$, then $r = \mathbf{v}^T\mathbf{v}$ is chi-square distributed with degree d. The probability density function (pdf), denoted by $f^{(d)}(r, \sigma)$, and the cumulated distribution function (cdf),

denoted by $F^{(d)}(r,\sigma)$, of r are

$$f^{(d)}(r,\sigma) = \begin{cases} \frac{1}{2^{d/2}\Gamma(d/2)}\left(\frac{r}{\sigma^2}\right)^{d/2-1} e^{-r/\sigma^2}, & \text{for } r > 0 \\ 0, & \text{for } r \leq 0 \end{cases} \tag{3.19}$$

$$F^{(d)}(r,\sigma) = \frac{\gamma\left(\frac{d}{2}, \frac{r}{2\sigma^2}\right)}{\Gamma(d/2)}. \tag{3.20}$$

where $\Gamma(x)$ is the gamma function

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-x}dt \tag{3.21}$$

and $\gamma(a,x)$ is the lower incomplete Gamma function

$$\gamma(a,x) = \int_0^x t^{a-1}e^{-t}dt. \tag{3.22}$$

From the Gaussian channel model, **Corollary 3.1** and **Corollary 3.2**, we know that $\|\mathbf{q} - R\mathbf{s}\|^2$ is chi-square distributed with degree n, and the PED of the ML path $\mathbf{s}_{ML}^{(i)}$ is a chi-square random variable of degree $d = n - i + 1$. When σ is given, the minimum $C^{(i)}$ is the inverse of $F^{(d)}(\epsilon^{(i)})$. That is, it can be derived by finding the $C^{(i)}$ which satisfies

$$F^{(d)}\left(C^{(i)}, \frac{\sigma_v}{2}\right) > 1 - \epsilon^{(i)}, \tag{3.23}$$

for $i = 1, 2, \ldots, n$.

The radius $C^{(i)}$ obtained from (3.23) for the constraint (3.18) requires the knowledge of the noise variance σ^2. However, it is difficult to acquire this value during decoding, and real-time comput the radii $C^{(1)}, C^{(2)}, \ldots, C^{(n)}$ results in huge computation overheads. Thus,

not only the error tolerance $\epsilon^{(i)}$'s, but σ^2 should be treated as a design parameter as well. For example, σ^2 can be selected according to SNR_{min}, defined as the SNR value where ML detection achieves some specific error performance. When the received SNR is lower than SNR_{min}, the transmit information is usually severely impaired and irrecoverable. Thus, only the SNR above SNR_{min} should be concerned. Consequently, σ^2_{max}, the noise variance corresponding to SNR_{min}, is regarded as an upper bound of the σ^2 in (3.23). As Figure 3.1 shows, the value $C^{(i)}$ increases with σ^2 for some fixed ϵ. Replacing σ^2 with σ^2_{max}, we can determine the radii at design time and remains constants during decoding. Equivalently, these radii provide looser radius constraints without effect on the error performance, but could lead to computation complexity increase since more paths are retained.

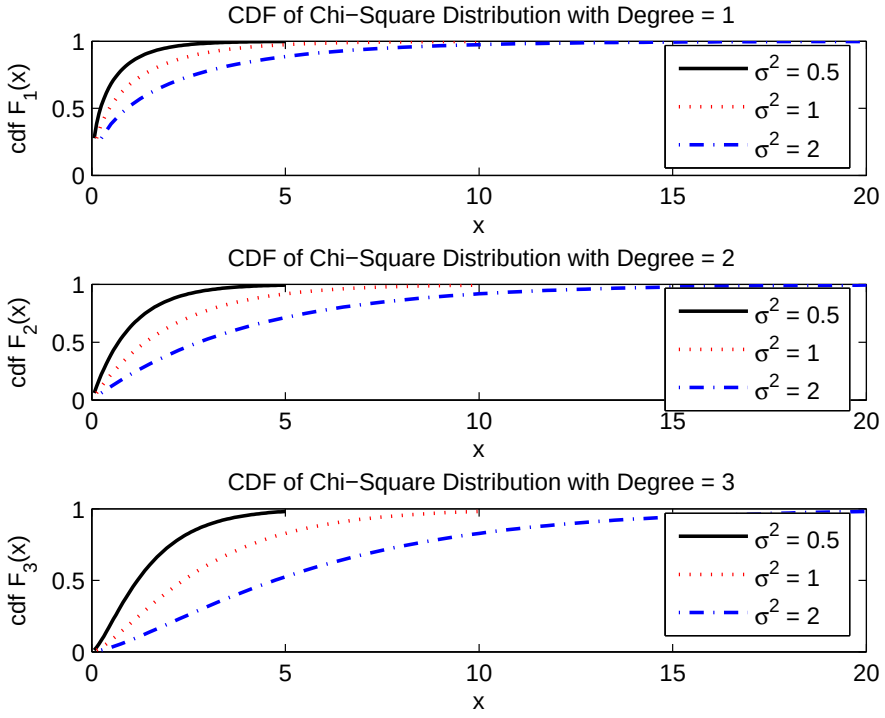


Figure 3.1: CDF of chi-square distribution of different degree of freedom and various σ^2.

We have presented the approach to obtain the radii, ie. the upper bounds, of the PEDs

at each decoding layer. The derivation is based on the Gaussian noise assumption and the path metric is computed by Euclidean distances. When the path metric is defined differently, the same constraint (3.18) still applies as long as the cdf of the path metric is given.

Replacing the path metric $T(\mathbf{s}^{(i)})$ defined in (3.7) by taking the absolute value is one common simplification in hardware implementation. That is, (3.7) is simplified to

$$\begin{aligned} T(\mathbf{s}^{(i)}) &= \sum_{i'=i}^{n} \left| q_{i'} - \sum_{j=i'}^{n} R_{i',j} s_j^{(i')} \right| \\ &= \sum_{i'=i+1}^{n} \left| q_{i'} - \sum_{j=i'}^{2N_t} R_{i',j} s_j^{(i')} \right| + \left| q_i - \sum_{j=i}^{n} R_{i,j} s_j^{(i)} \right| \\ &= T(\mathbf{s}^{(i+1)}) + e(\mathbf{s}^{(i)}). \end{aligned} \tag{3.24}$$

In the following, derivation of the radii for the path metric (3.24) will be described by an example.

Corollary 3.3. Given a random variable X with pdf $g_X(x)$, the pdf of $|X|$ can be derived by

$$g_{|x|}(x) = \begin{cases} g_X(x) + g_X(-x), & \text{for } x \geq 0; \\ 0, & \text{for } x < 0. \end{cases} \tag{3.25}$$

Corollary 3.4. For i.i.d. randmon variables $v_1, v_2, \ldots, v_d$ with pdf f_i, the pdf of $\sum_{i=1}^{d} v_i$, denoted by f_s , can be derived by

$$f_s = f_1 \otimes f_2 \otimes \cdots \otimes f_d, \tag{3.26}$$

where the operator $\otimes$ represents linear convolution.

From **Corollary3.3**, the pdf of $|v_i| = |q_i - \sum_{j=i}^{n} R_{i,j} s_j^{(i)}|$, denoted by $\hat{g}(v)$, should be

$$\hat{g}(v) = \begin{cases} \frac{2}{\sqrt{2\pi}\sigma} \exp(\frac{-v^2}{2\sigma^2}), & \text{for } v > 0, \\ 0, & \text{for } v \leq 0; \end{cases} \tag{3.27}$$

for $v_i = q_i - \sum_{j=i}^{n} R_{i,j} s_j^{(i)}$ is Gaussian distributed. Let $f^{(1)}(v)$ be the pdf of $T(\mathbf{s}_{ML}^{(n)})$. Since $T(\mathbf{s}_{ML}^{(n)}) = |q_n - R_{n,n} s_{ML}^{(n)}| = |v_n|$, $f^{(1)}(v)$ equals to $\hat{g}(v)$ defined in (3.27). Following **Corollary3.4**, $f^{(d)}(v)$, the pdf of $T(\mathbf{s}_{ML}^{(i)}) = |v_i| + T(\mathbf{s}_{ML}^{(i+1)})$, can be derived recursively from

$$f^{(d)}(v) = f^{(d-1)}(v) \otimes f^{(1)}(v) \tag{3.28}$$

for $d = n - i + 1$.

The aforementioned approach to derive the radii can be applied to other variations of path metric as long as the recursive form (3.7) or (3.24) holds. Similarly, the radii can be determined at design time and independent of the channel.

3.2.2 Multi-K-Best Algorithm with Radius Constraint

The radius constraints introduced in **Section 3.2.1** allow the decoder to prune less likely paths before it proceeds the computation of the next layer. Similar to the depth-first decoders, the computation also varies. To maintain predictable, manageable complexity, and decoding speed, a maximum number of the retained paths at each layer should be set.

The combination of the radius constraints and K-best algorithm brings on an adequate approach benefiting from both depth-first and breadth-first search strategies. At each layer, the decoder first keeps all paths satisfying the radius constraint. Only the K paths corresponding to the K smallest PEDs are preserved if the number of the retained paths exceeds

K. Similar to K-best algorithm, a sorter is required to distinguish the K best paths, and the sorting operation dominates the decoding computation complexity. If a constant K value is chosen for each decoding layer, the suitable K value can be easily obtained by simulation. If each decoding layer corresponds to a distinct K value, the resulted *multi-K-best algorithm* facilitates computationally efficient and high-throughput decoder designs. However, empirically deriving the multiple K values from a vast combinations is very time-consuming and almost infeasible.

Instead of determining the multiple K's by simulation, we analyze the expected number of paths retained by the radius constraint of each layer, and the multiple K values can be set according to the expected retained path number.

Let $\mathbf{s}_a^{(i)}$ be the *ambiguous path* of the i-th layer that also satisfies the radius constraints:

$$\begin{aligned} T(\mathbf{s}_a^{(n)}) &\leq C^{(n)} \\ T(\mathbf{s}_a^{(n-1)}) &\leq C^{(n-1)} \\ &\vdots \\ T(\mathbf{s}_a^{(i)}) &\leq C^{(i)}. \end{aligned} \tag{3.29}$$

Let $\xi^{(i)} = T(\mathbf{s}_a^{(i)}) - T(\mathbf{s}_a^{(i+1)})$ denote the path increment of $\mathbf{s}_a$ from layer-$(i+1)$ to layer-i, and the following increment constraints must hold:

$$\begin{aligned} \xi^{(n)} = T(\mathbf{s}_a^{(n)}) &\leq C^{(n)} \\ \xi^{(n-1)} = T(\mathbf{s}_a^{(n-1)}) - T(\mathbf{s}_a^{(n)}) &\leq C^{(n-1)} - \alpha^{(n-1)} C^{(n)} \\ &\cdots \\ \xi^{(i)} = T(\mathbf{s}_a^{(i+1)}) - T(\mathbf{s}_a^{(i)}) &\leq C^{(i)} - \alpha^{(i)} C^{(i+1)} \end{aligned} \tag{3.30}$$

for $0 < \alpha^{(i)} \leq 1$. Furthermore, $\mathbf{s}_a^{(i)}$ is $\|\mathbf{\Delta}^{(i)}\|^2$ away from the ML path $\mathbf{s}_{ML}^{(i)}$. That is,

$$\mathbf{s}_a^{(i)} = \mathbf{s}_{ML}^{(i)} - \mathbf{\Delta}^{(i)}, \tag{3.31}$$

where $\Delta^{(i)} = [\Delta_i^{(i)}, \Delta_{i-1}^{(i)}, \ldots, \Delta_n^{(i)}]^T$ and $\Delta_j \in \{\pm\delta, \pm 2\delta, \ldots, \pm(M-1)\delta\}$ for M-PAM signal mapping. Thus, the path matric of $\mathbf{s}_a^{(i)}$ is

$$\begin{aligned} T(\mathbf{s}_a^{(i)}) &= \|\mathbf{q}^{(i)} - \mathbf{R}^{(i)} s_a^{(i)}\|^2 \\ &= \|\mathbf{q}^{(i)} - \mathbf{R}^{(i)} s_{ML}^{(i)} + \mathbf{R}^{(i)} \Delta^{(i)}\|^2 \\ &= \|\mathbf{v}^{(i)} + \mathbf{R}^{(i)} \Delta^{(i)}\|^2, \end{aligned} \tag{3.32}$$

where $\mathbf{R}^{(i)}$ represents the last i-th rows of the channel matrix $\mathbf{R}$. Note that

$$\xi^{(i)} = \left(v_i + \sum_{j=i}^{n} R_{i,j} \Delta_j^{(i)} \right)^2 = \eta_i^2 \tag{3.33}$$

which is also chi-square distributed with degree of freedom 1 and

$$\eta_i \sim \mathcal{N}\left(0, \frac{1}{2}\left(\sigma^2 + \sum_{j=i}^{n} (\Delta_j^{(i)})^2\right)\right) \tag{3.34}$$

is a zero-mean Gaussian variable of variance $\frac{1}{2}\left(\sigma^2 + \sum_{j=i}^{n} (\Delta_j^{(i)})^2\right) = \frac{1}{2}(\sigma^2 + \|\mathbf{\Delta}^{(i)}\|^2)$. Let us define $\lambda^{(i)} \triangleq \frac{1}{\delta^2} \sum_{j=i}^{n} (\Delta_j^{(i)})^2$ which can be conducted to the recursive form

$$\lambda^{(i)} = \lambda^{(i+1)} + \frac{(\Delta_i^{(i)})^2}{\delta^2}. \tag{3.35}$$

Let $F_\Xi(\xi^{(i)}|\lambda^{(i)})$ be the cdf of $\xi^{(i)}$ for $\lambda^{(i)}$ is given. Figure 3.2 is an illustrative example of

$F_{\Xi}(C^{(i)} - \alpha^{(i)}C^{(i+1)}|\lambda^{(i)})$ for $i = 7$ of a 4×4 64-QAM system at SNR = 25dB, where the noise variance σ^2 is 0.0032 and the constellation spacing δ is 0.0119. Note that $F_{\Xi}(C^{(i)} - \alpha^{(i)}C^{(i+1)}|\lambda^{(i)})$ decreases as $\lambda^{(i)}$ increases. In other wards, a farther $\mathbf{s}_a^{(i)}$ from $\mathbf{s}_{ML}^{(i)}$ has lower probability satisfying the radius constraints.

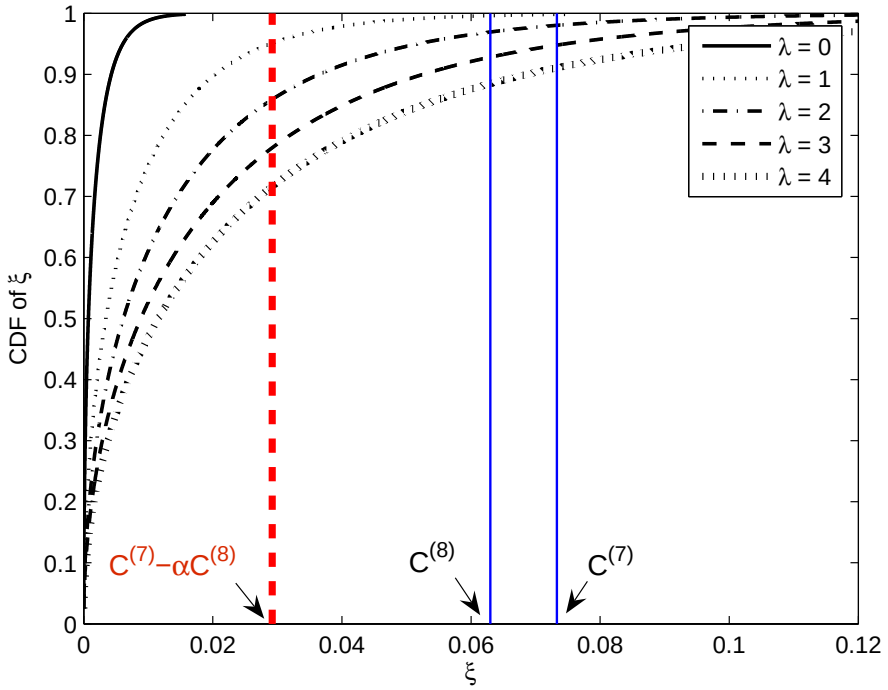


Figure 3.2: CDF of the χ^2 variable $\xi(\mathbf{\Delta}^{(i)})$ given various $\lambda = \frac{1}{\delta^2}\sum_{j=i}^{n}(\Delta_j^{(i)})^2$

Moreover, since all the elements in $\mathbf{R}$ are i.i.d. Gassian random variables, η_i and η_j are also independent Gaussian random variables for all $i \neq j$. It follows that $\xi^{(i)}$ and $\xi^{(j)}$ are

independent chi-square random variables. Therefore,

$$\begin{aligned} & Pr\left(T(\mathbf{s}_a^{(i)}) \le C^{(i)}, T(\mathbf{s}_a^{(i+1)}) \le C^{(i+1)}, \ldots, T(\mathbf{s}_a^{(n)}) \le C^{(n)} \mid \lambda^{(i)}\right) \\ = \; & Pr\left(\xi^{(i)} \le C^{(i)} - \alpha^{(i)} C^{(i+1)}, \xi^{(i+1)} \le C^{(i+1)} - \alpha^{(i)} C^{(i+2)}, \ldots, \xi^{(n)} \le C^{(n)} \mid \lambda^{(i)}\right) \\ = \; & \prod_{j=i}^{n} F_{\Xi}\left(C^{(j)} - \alpha^{(j)} C^{(j+1)} \mid \lambda^{(i)}\right). \end{aligned} \tag{3.36}$$

Furthermore, the independency among the transmitted symbols $s_1, s_2, \ldots, s_n$ leads to

$$Pr\left(\mathbf{\Delta}^{(i)} = \mathbf{a}\right) = \prod_{j=i}^{n} Pr\left(\Delta_j^{(i)} = a_j^2\right) \tag{3.37}$$

for $a_j \in \{0, \pm\delta, \pm 2\delta, \ldots, \pm(M-1)\delta\}$. Besides, each $\mathbf{\Delta}^{(i)}$ corresponds to a dinstint $\mathbf{s}^{(i)}$ over all M^{n-i+1} possible points in the $(n-i+1)$-dimensional sphere. With equal prior probability assumption, we can have $Pr\left(\mathbf{\Delta}^{(i)} = \mathbf{a}\right) = \frac{1}{M^{n-i+1}}$ and then

$$\begin{aligned} Pr\left(\lambda^{(i)} = \lambda\right) &= \begin{pmatrix} \lambda \\ m_i^2, m_{i+1}^2, \ldots, m_n^2 \end{pmatrix} Pr\left(\mathbf{\Delta}^{(i)} = \mathbf{a}\right) \\ &= \begin{pmatrix} \lambda \\ m_i^2, m_{i+1}^2, \ldots, m_n^2 \end{pmatrix} \frac{1}{M^{n-i+1}}, \end{aligned} \tag{3.38}$$

where the integers

$$m_j = \frac{a_j}{\delta} \tag{3.39}$$

for $j = 1, 2, \ldots, n$ and $\begin{pmatrix} \lambda \\ m_i^2, m_{i+1}^2, \ldots, m_n^2 \end{pmatrix}$ denotes the number of distinct $\{m_i^2, m_{i+1}^2, \ldots, m_n^2\}$ resulting in $\lambda = \sum_{j=i}^{n} m_j^2$,

Subsequently, let $K^{(i)}$ be the maximum number of retained paths of the i-th layer, and

$\bar{N}^{(i)}$ be the average number of $\mathbf{s}^{(i)}$ satisfying the increment constraints in (3.30). By definition, $\bar{N}^{(i)}$ can be expressed by

$$\begin{aligned}\bar{N}^{(i)} &= M^{n-i+1} Pr\left(T(\mathbf{s}^{(i)}) \le C^{(i)}, T(\mathbf{s}^{(i+1)}) \le C^{(i+1)}, \ldots, T(\mathbf{s}^{(n)}) \le C^{(n)}\right) \\ &= M^{n-i+1} \sum_{\lambda} Pr\left(\lambda^{(i)} = \lambda\right) \\ &\quad \times Pr\left(T(\mathbf{s}^{(i)}) \le C^{(i)}, T(\mathbf{s}^{(i+1)}) \le C^{(i+1)}, \ldots, T(\mathbf{s}^{(n)}) \le C^{(n)} | \lambda^{(i)} = \lambda\right). \quad (3.40)\end{aligned}$$

From(3.36)and (3.38), (3.40) becomes

$$\begin{aligned}\bar{N}^{(i)} &= \sum_{m_i^2, m_{i+1}^2, \ldots, m_n^2} \binom{\lambda}{m_i^2, m_{i+1}^2, \ldots, m_n^2} \prod_{j=i}^{n} F_{\Xi}\left(C^{(j)} - \alpha^{(j)} C^{(j+1)} \middle| \lambda^{(j)} = \sum_{j'=j}^{n} m_{j'}^2\right) \\ &= \sum_{m_i=0}^{M-1} \sum_{m_{i+1}=0}^{M-1} \cdots \sum_{m_n=0}^{M-1} \prod_{j=i}^{n} F_{\Xi}\left(C^{(j)} - \alpha^{(j)} C^{(j+1)} \middle| \lambda^{(j)} = \sum_{j'=j}^{n} m_{j'}^2\right), \quad (3.41)\end{aligned}$$

and $K^{(i)}$ will be determined as a function of $\bar{N}^{(i)}$.

$$K^{(i)} = \left\lceil \beta \bar{N}^{(i)} \right\rceil \quad (3.42)$$

could be one simpliest form; the function $\lceil x \rceil$ returns the smallest integer that is greater than or equal to x.

The goal of employing multi-K-best algorithm is to reduced the complexity of the K-best algorithm while remaining similar error performance. Thus we can confine β so that $\max\{K^{(i)} | i = 1, 2, \ldots, n\} \le K$. Moreover, the number of preserved paths decreases with i because there are less paths that meets all the radius constraints from layer n to layer i. An

intuitive guess of the $K^{(i)}$'s could be

$$K^{(i)} < K, \qquad \text{when } i \text{ is small;}$$
$$K^{(i)} \approx K, \quad \text{when } i \text{ is close to } n.$$

The $K^{(i)}$'s can be determined by

$$K^{(i)} = \begin{cases} K, & \text{if } i \geq n - n_K \\ \left\lceil \beta \bar{N}^{(i)} \right\rceil, & \text{if } i < n - n_K. \end{cases} \tag{3.43}$$

for $\beta > 0$, $0 \leq n_K \leq n$ and K is parameter for the conventional K-best algorithm. Note that β is a tradeoff between complexity and performance. Now the problem of finding a set of suitable n-dimensional K values is reduced to searching for a suitable 1-dimensional β factor, which can be easily derived empirically.

Note that $\bar{N}^{(i)}$ derived in (3.41) is dependent of $\alpha^{(i)}, \alpha^{(i+1)}, \ldots, \alpha^{(n)}$, which are difficult to be obtained. Thus we approximate all the $\alpha^{(i)}$ by a constant α for $i = 1, 2, \ldots, n$. It can be observed in Figure 3.2 that the cdf $F_{\Xi}(C^{(i)} - \alpha C^{(i+1)} | \lambda)$ is a non-increasing function of α. Larger α equivalently provides a smaller estimate of $\bar{N}^{(i)}$; smaller α results to over-estimate of $\bar{N}^{(i)}$, and $0 < \alpha \leq 1$. Fig.3.3 illustrates the expected retained path $\bar{N}^{(i)}$ for $n = 8$ when approximated by a constant α. It is observed that α is comparatively less related to $\bar{N}^{(i)}$ for low SNR scenarios, whereas for higher SNR environments $\bar{N}^{(i)}$ explodes when α is too small; consequently, little information can be delivered. According to the results in Fig.3.3, a reasonable guess of α for the SNR above 20dB can be $0.9 < \alpha \leq 1.0$.

Approximated by $\alpha = 1.0$, the expected number of retained paths $\bar{N}^{(i)}$ for $n = 8$ is shown in Fig.3.4. It is perceived that $\bar{N}^{(i)}$ increases exponentially with the dimension, which

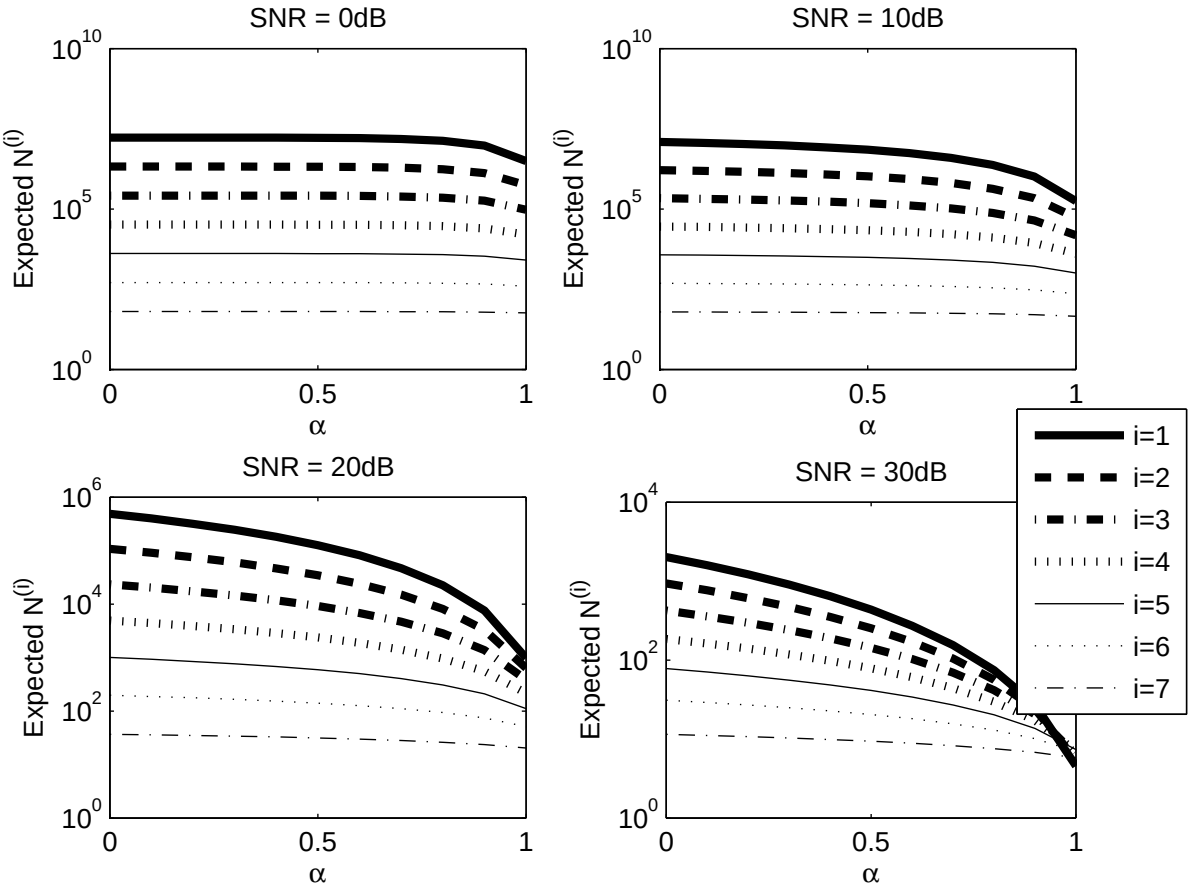


Figure 3.3: The expected retained path $\bar{N}^{(i)}$ for $n = 8$ and $0 < \alpha^{(i)} \leq 1$.

is $n - i + 1$, of each decoding layer for low SNR values. For higher SNR, $\bar{N}^{(i)}$ approaches to some constant values, inferring that the early-pruning technique can provide significant reduction in computation complexity when received signal strength is high.

For other path metric definitions, the same analysis techniques can be applied. By modifying the distributions in (3.33) and (3.34), $K^{(i)}$'s and $\bar{N}^{(i)}$ can be derived.

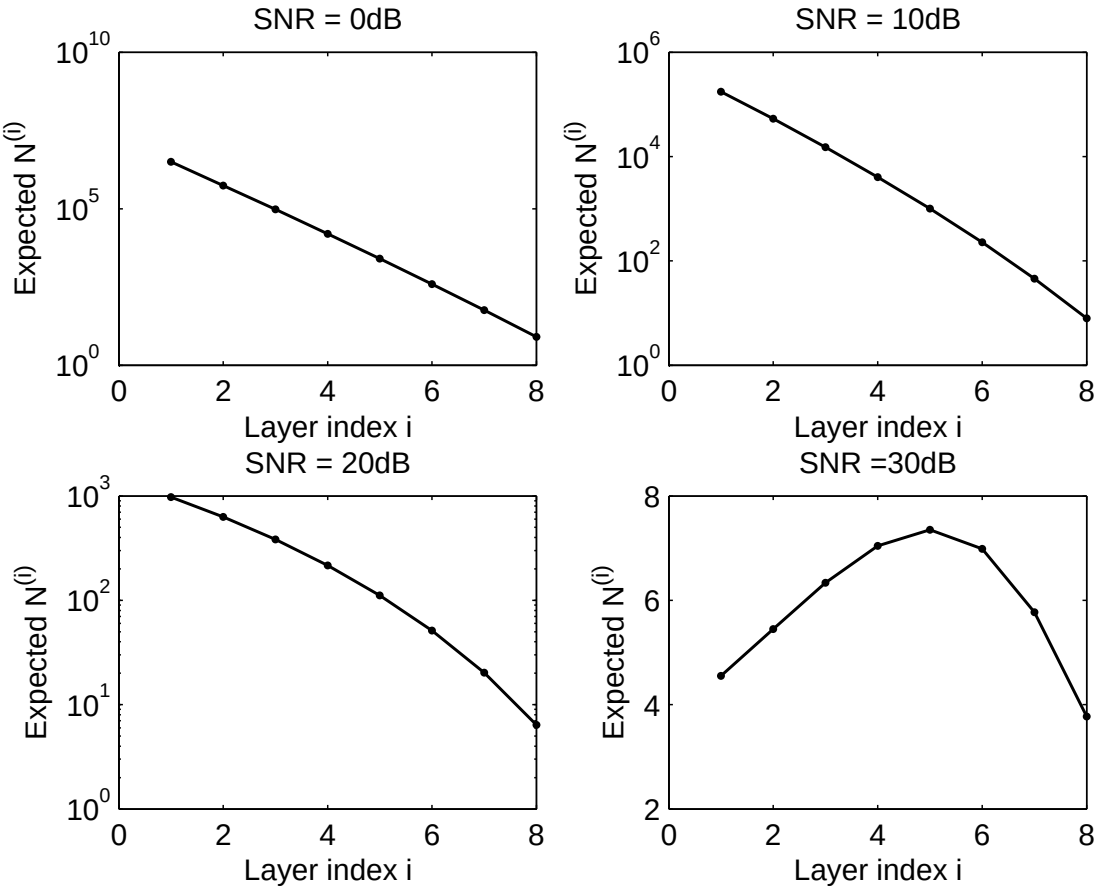


Figure 3.4: The expected number of retained paths $\bar{N}^{(i)}$ for $n = 8$ and $\alpha^{(i)} = 1.0$.

3.2.3 Coarse-Granularity Sorting

Whether K-best algorithm or the aforesaid multi-K-best algorithm, sorting operation always dominates the computation complexity. In fact, at each layer the decoder only requires the K *best* values, which means the order among the K best values is unnecessary. Therefore, if we can replace the strictly sorting by other approximately sorting schemes, the computation complexity can be greatly reduced.

Since the radius constraint equivalently reveals the range of the retained paths, a *coarse-*

granularity sorting strategy can be applied. First, the range of all the i-th layer path metrics, which is $(0, C^{(i)}]$, is partitioned into L regions, and an index l' is assigned to the path metric $T(\mathbf{s}^{(i)})$ by

$$l' = \begin{cases} l, & \text{if } \frac{l-1}{L}C^{(i)} < T(\mathbf{s}^{(i)}) \leq \frac{l}{L}C^{(i)}; \\ L+1, & \text{if } T(\mathbf{s}_k^{(i)}) > C^{(i)}. \end{cases} \tag{3.44}$$

Let k_i denotes the number of paths with index $l' = l$, the decoder first finds the minimum l_{max} such that $k_1 + k_2 + \cdots k_{l_{max}} > K$ and $k_1 + k_2 + \cdots + k_{l_{max}-1} < K$. The $k_1 + k_2 + \cdots k_{l_{max}-1}$ paths in region $(0, \frac{l_{max}-1}{L}C^{(i)}]$ are then selected and kept. Finally, the decoder randomly chooses $K - k_1 - k_2 - \cdots - k_{l_{max}-1}$ from region $(\frac{l_{max}-1}{L}C^{(i)}, \frac{l_{max}}{L}C^{(i)}]$. As a result, sorting can be approximated by a few comparators. Note that when there is no path satisfying the radius constraint, i.e, all the $l' = L + 1$, the decoder has to search for the path with minimum path metric. In this case, the number of retained path is 1. Figure 3.5 illustrates a $K = 6$ example, and the balls in the l-th bucket denote the paths path metrics within $(\frac{l_{max}-1}{L}C^{(i)}, \frac{l_{max}}{L}C^{(i)}]$.

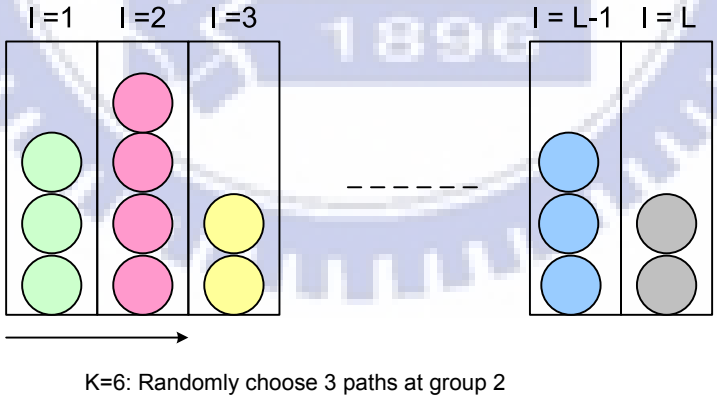


Figure 3.5: Coarse-granularity sorting for $K = 6$.

3.3 Complexity Analysis

The computation complexity of sphere decoders can be measured by the number of floating point operations, including multiplication, addition, and subtraction. For depth-first search, complexity is often evaluated by the term *expected complexity exponent* [61], which is defined as

$$E_C \triangleq \frac{\log Q(n, \sigma^2, C)}{\log n}, \tag{3.45}$$

where σ^2 is the noise variance and $Q(n, \sigma^2, C)$ represents the expected number of floating point operations corresponding to an n-dimensional hypersphere with radius C. By definition, $Q(n, \sigma^2, C)$ can be expressed by

$$\begin{aligned} Q(n, \sigma^2, C) \;\; = \;\; & (\text{expected number of nodes in the sphere}) \\ & \times (\text{number of floating point operations per node}). \end{aligned} \tag{3.46}$$

In [61], $E_c \approx \frac{n}{\log n}$ for large σ^2; when $\sigma^2 < 1$, E_c is almost constant for a wide range of n, leading to the polynomial complexity that is expected.

The expected complexity exponent for breadth-first decoders can be defined in the same manner. For conventional K-best algorithm that possesses constant computation, the expected complexity exponent is independent of σ^2 and

$$E_{C_K} \approx \frac{\log \left(K \sum_{i=1}^{n} 2(n - i + M) \right)}{\log n}, \tag{3.47}$$

which approaches to $\frac{\log K}{\log n} + 2$ as $n \gg M$. Similarly, E_c of multi-K-best algorithm is

$$E_{C_{MK}} \approx \frac{\log \left(\sum_{i=1}^{n} K^{(i)} 2(n - i + M) \right)}{\log n}. \tag{3.48}$$

Applying the radius constraints, the expected complexity expopnent of the aforementioned early-pruned breadth-first sphere decoding algorithm will be upper-bounded by (3.47) or (3.48). Similar to the depth-first searching strategy, the complexity depends on the noise variance σ^2.

3.3.1 Expected Complexity Exponent

By definition, the $Q(n, \sigma^2, C)$ in (3.46) of the proposed early-pruned breadth-first sphere decoder will be modified to

$$Q(n, \sigma^2, \epsilon) = \sum_{i=1}^{n} \bar{N}^{(i)}(\epsilon^{(i)}) 2(n - i + M), \tag{3.49}$$

where $\epsilon = [\epsilon^{(1)}, \epsilon^{(2)}, \ldots, \epsilon^{(n)}]^T$ determines the radii $\{C^{(1)}, C^{(2)}, \ldots, C^{(n)}\}$ and $\bar{N}^{(i)}(\epsilon^{(i)})$ is the resulted expected number of nodes in the $(n-i+1)$-dimensional sphere of radius $C^{(i)}$. The expected complexity excponent, denoted by $E_{C_{EP}}$, is

$$E_{C_{EP}} \approx \frac{\log\left(\sum_{i=1}^{n} \bar{N}^{(i)}(\epsilon^{(i)}, \alpha) 2(n - i + M)\right)}{\log n}. \tag{3.50}$$

Fig.3.6 illustrates the E_C versus the sphere of degree n for the 8-PAM signal mapping where the radii $C^{(1)}, C^{(2)}, \ldots, C^{(8)}$ are determined by $\epsilon^{(1)} = \epsilon^{(2)} = \cdots = \epsilon^{(8)} = 0.0001$ and $\alpha^{(1)} = \alpha^{(2)} = \cdots = \alpha^{(8)} = 1.0$. Besides, the E_C of the conventional 64-best algorithm is illustrated for reference. It is observed that the expected complexity exponent increases with the degree n when SNR is small. For larger SNR condition (SNR = 30dB in this example), E_C tends to be a constant. Moreover, the E_c of 64-best algorithm approaches to some constant as well, indicating that the complexity of the two algorithms are both polynomial with n in high SNR scenarios. Besides, the smaller E_c shows the early-pruning

scheme results in lower average computation complexity as compared to 64-best algorithm.

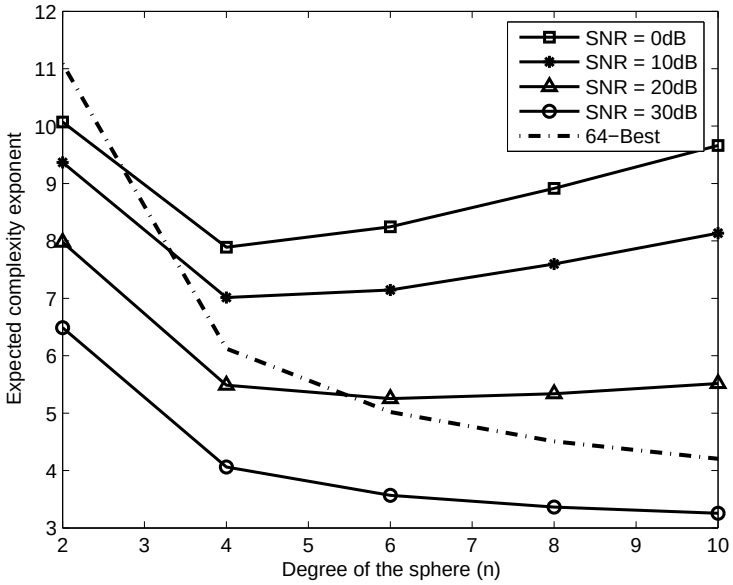


Figure 3.6: Expected complexity exponents (Ec) of early-pruned breadth-first sphere decoders.

3.3.2 Expected Computation Complexity

The concept of the expected complexity exponent was first introduced for depth-first sphere decoders, and only the complexity of addition and multiplication are evalutated. In fact, sorting complexity, which dominates the computation of breadth-first decoders, should be considered as well.

Comparison is the basic operation for sorting, among several sorting algorithms, the number of comparisons executed by an N-input sorter ranges from $N \log_2 N$ to N^2. In the best case, it takes about

$$\sum_{i=1}^{n} MK^{(i)} \log_2(MK^{(i)}) \tag{3.51}$$

comparisons to decode one n-dimensional M-PAM mapped signal by the multi-K-best algorithm. Replaced by the coarse-granularity sorting introduced in previous section, the sorter requires about $K^{(i)}M \times L$ comparisons at the i-th layer, and the comparision number can be reduced to $K^{(i)}M \log_2 L$ if binary-search is employed.

For strictly sorters, the sorting complexity of the early-pruned multi-K-best algorithm is about

$$\sum_{i=1}^{n-1} \left(P_{K^{(i)}} \bar{N}^{(i)}(\epsilon^{(i)})M + (1 - P_{K^{(i)}})K^{(i)}M \log_2(K^{(i)}M) \right), \tag{3.52}$$

where $P_{K^{(i)}}$ is the probabililty that the number of paths satisfying the radius constraint at layer-i is less than $K^{(i)}$ and no sorting is required. For coarse-granularity sorting approach, the sorting complexity will be reduced to a linear function of $\bar{N}^{(i)}(\epsilon^{(i)})$, which is

$$M \left(\sum_{i=1}^{n-1} \bar{N}^{(i+1)}(\epsilon^{(i+1)}) + 1 \right) + L \sum_{i=1}^{n} \bar{N}^{(i)}(\epsilon^{(i)}) \tag{3.53}$$

or

$$M \left(\sum_{i=1}^{n-1} \bar{N}^{(i+1)}(\epsilon^{(i+1)}) + 1 \right) + (\log_2 L) \sum_{i=1}^{n} \bar{N}^{(i)}(\epsilon^{(i)}). \tag{3.54}$$

Note that the first term in (3.53) and (3.54) is contributed by checking the radius constraints, and the second term is resulted from the coarse-granularity sorting.

Furthermore, due to the regular computation of the path metric and the breadth-first search nature, all the operations performed by the early-pruned sphere decoder can be easily predicted by $\bar{N}^{(i)}(\epsilon^{(i)})$, providing more explicit complexity analysis. That is, the number of additions and multiplications performed by early-pruned sphere decoder can be estimated by

$$\sum_{i=1}^{n} \bar{N}^{(i)}(\epsilon^{(i)})(n - i + M). \tag{3.55}$$

3.4 Simulation Results

A 4×4 MIMO system was simulated. Random binary data with equal prior probability was mapped by 64-QAM signaling and transmitted in an uncorrelated flat fading plus AWGN channel. The results of error performance and average computation complexity are given in the following.

3.4.1 Error Performance

Figure 3.7 and Figure 3.8 illustrate the symbol error rate and bit error rate of several detection schemes. The ML detection is realized by Schnorr-Euchnorr sphere decoding algorithm. All the $C^{(i)}$ are derived with $\epsilon^{(i)} = 0.0001$ at minimum working SNR as 25 dB.

First, we can observe obvious diversity gain provided by the ML detection as compared to the zero-forcing approach. Moreover, degradation occurs when the K value of the K-best algorithm is small. It is perceived that K should be greater than 32 for this system, and the error performance of the 64-best algorithm is nearly the same as the ML detection. Thus, $K = 64$ is chosen for the early-pruned K-best algorithm, which is represented by EP-64-best henceforth. Accordingly, with $\alpha = 1.0$, $n_K = 3$, and $\beta = 4.5$; the resulted $K^{(i)}$'s for the multi-K-best algorithm are derived as 21, 25, 29, 32, 34, 64, 64, 8, from the first to the eighth layer. Furthermore, the sorting operation of both EP-64-best and EP-multi-K-best are realized by the coarse-granularity sorting where the range $(0, C^{(i)}]$ is partitioned into 16 regions, i.e., $L = 16$.

As it is shown in these two figures, the degradation resulted from the three schemes (early-pruned, multi-K, and coarse-granularity sort) is hardly recognized for SER and BER above 10^{-5}. In fact, it will be shown in subsequent that the computation complexity can be

greatly reduced.

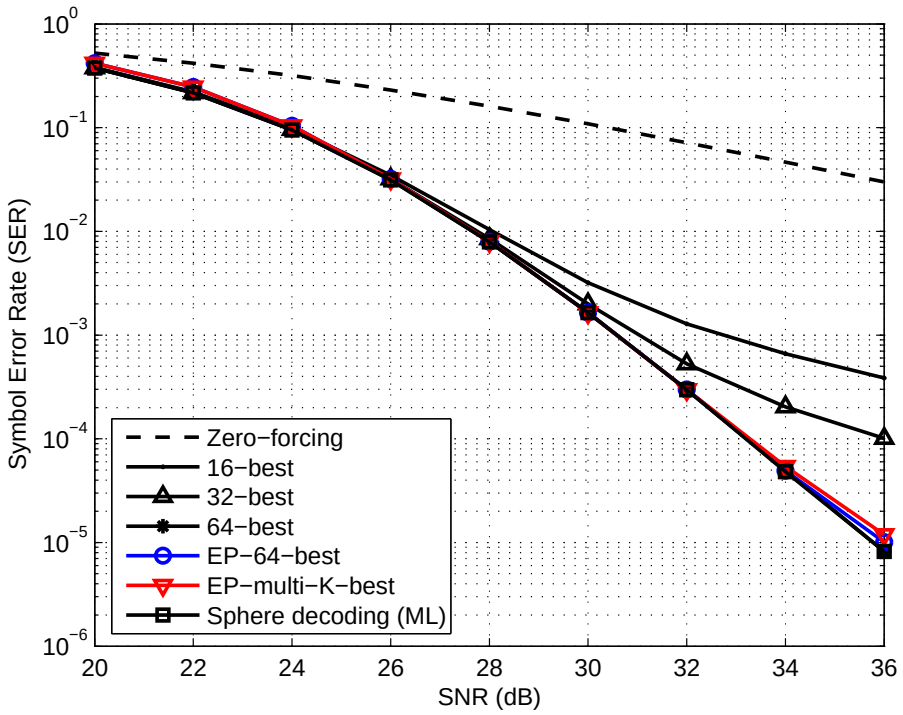


Figure 3.7: Symbol error rate of 4×4 64-QAM MIMO system.

3.4.2 Computation Complexity

The computation complexity of a sphere decoder is determined by the number of nodes visited during the tree-search process. Fig.3.9 illustrates the simulated probability of the early-pruned sphere decoder combined with 64-best algorithm (EP-64-best). The probability of k paths retained is truncated at $k = 64$, wherein the spikes shown in the four subfigures of Fig.3.9. Moreover, for $k > 20$ in each of the four subfigures the probability becomes small, which is about 10^{-3} or smaller. The low probability reveals that the average number of retained paths should be very small. Moreover, since the radii are derived by setting

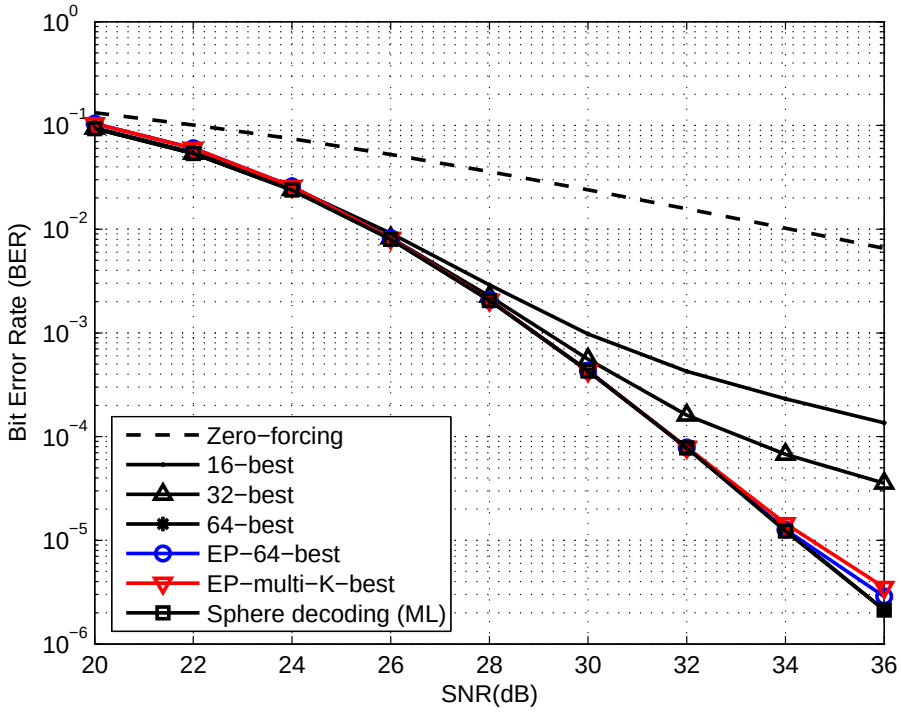


Figure 3.8: Bit error rate of 4×4 64-QAM MIMO system.

the minimum working SNR as 25dB, the shapes of the simulated probabilities tend to be sharper for SNR lower than 25dB for the radii becomes too strict. Similarly, the shapes of the probabilities become wider for SNR higher than 25dB since the radii becomes a looser restriction. Furthermore, we can observe that the probabilities corresponding to smaller layer index i is sharper. This can be explained by regarding the early-pruning scheme as a filter that filters out the less likely paths at each layer. The decoder proceeds from the 8-th layer to the first layer. As a results, the number of paths satisfying all the radius constraints becomes fewer when i is small. Thus it is perceived in the figures that the probability corresponding to $i = 1$ has the sharpest shape for all SNR values.

Similar phenomenon can be observed in Figure 3.10, which is the simulated probability of

the early-pruned sphere decoder combined multi-K best algorithm (EP-multi-K-best) where the $K^{(i)}$ for multi-K algorithm is 21, 25, 29, 32, 34, 64, 64, 8, from the first to the eighth layer.

Figure 3.11 and Figure 3.12 present the average number of retained paths at each layer for EP-64-best and EP-multi-K-best for different SNR values. The solid lines in both figures are the expected number of paths derived by (3.41) with SNR equals to 30dB, $\alpha = 1.0$, and $\epsilon^{(i)} = 0.0001$. First, let us compare the results corresponding to SNR = 30dB. The expected values are derived based on the radius constraints only, and the simulated average values are derived with additional 64-best or multi-K-best restrictions. Thus the values derived from simulation should be smaller than the theoretically derived values. The two figures show that (3.41) can give a tight upper bound in estimating the expected number of retained paths. Furthermore, the simulation results also shows that setting $\alpha^{(i)} = \alpha = 1.0$ can provide a quite accurate approximation in computing (3.41).

Next, let us examine the results of SNR at 26dB and 30dB, which are higher than the minimum working SNR (25dB). Since the radii become even looser constraints for the case of 30dB, the number of retained paths is a little larger as compared to the results of SNR at 26dB. Similarly, the radii are stricter restrictions for lower SNR values. As a result, it is perceived in the two figures that the case of SNR at 16dB always has the smallest value for the same layer index i.

Table 3.1 shows the average number of operations performed for SNR at 30dB. Conventional 64-best algorithm is presented as a reference, to which the results of other schemes are normalized. It is perceived that by early-pruning and coarse-granularity sorting, more than 90% of the computations, which are comparisons, multiplications, and additions , can be saved. Furthermore, it can be observed in Figure 3.7 and Figure 3.8 that the degradation

Table 3.1: Computation complexity

Operation/Method	Conventional 64-best	EP (Theoretical) ($L = 16$)	EP-64-best ($L = 16$)	EP-multi-K-best ($L = 16$)
Comparison	27648 (100%)	545 (1.98%)	425 (1.55%)	401 (1.46%)
Multiplication	5440 (100%)	545 (10.02%)	415 (7.63%)	390 (7.17%)
Addition	5440 (100%)	545 (10.02%)	415 (7.63%)	390 (7.17%)

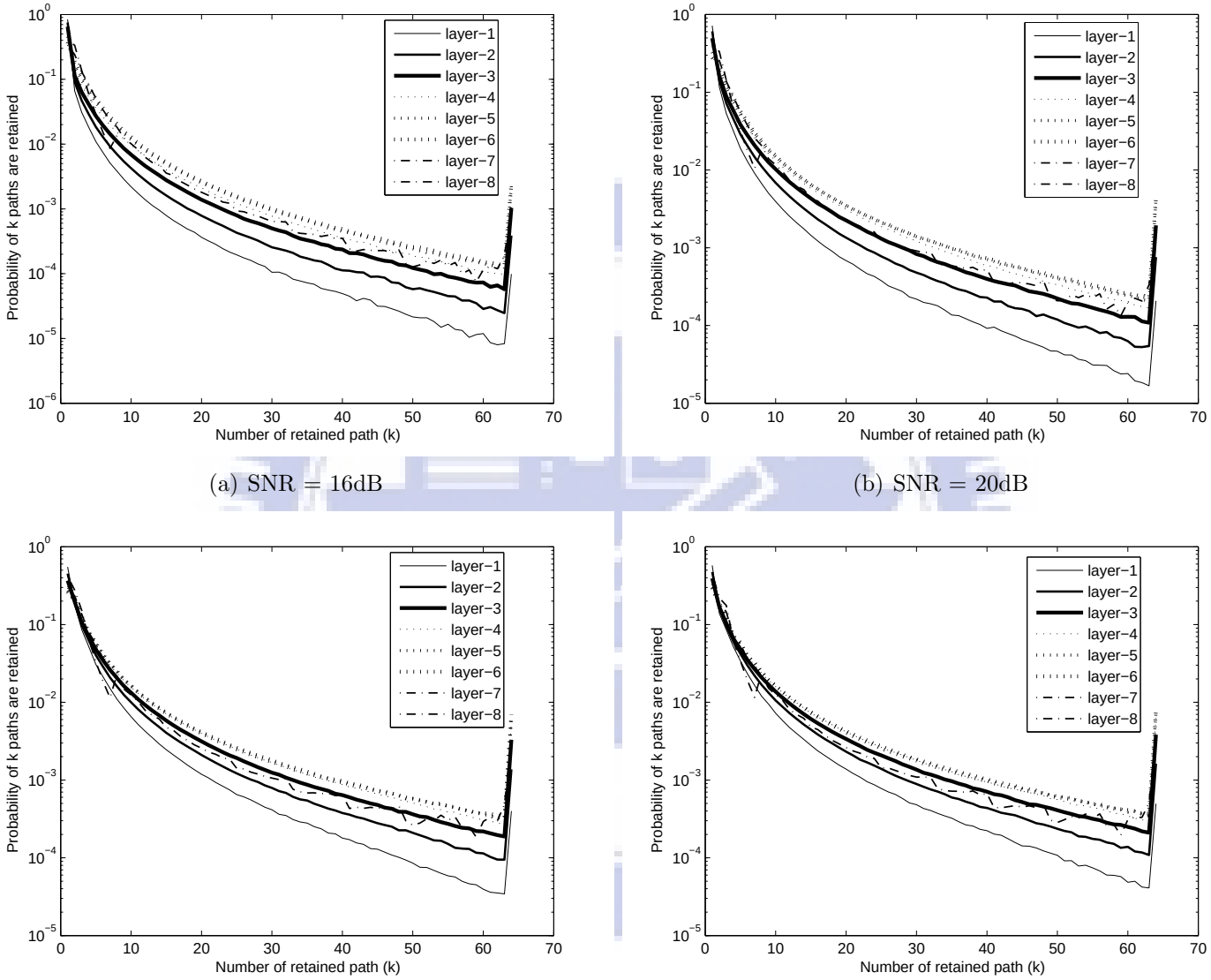
at this SNR value is nearly imperceptible. Besides, although complexity reduction shown in Table 3.1 from 64-best algorithm to multi-K-best algorithm is not significant, the benefit will become more obvious in hardware implementation. On average, the number of paths satisfying the radius constraints is far less than 64, therefore the average decoder throughputs of EP-64-best and EP-multi-K-best are similar. However, when the path number exceeds K, the decoder reaches its lowest decoding speed. Since multi-K best with $K^{(i)}$ smaller than 34 for $i < 6$ in this case, the corresponding worst decoding speed can be nearly doubled as compared to EP-64-best algorithm.

3.5 Summary

In this chapter, early-pruning technique for breadth-first sphere decoders are proposed. A set of distinct radii can be derived theoretically based on the error tolerance and the received data statistics. Combining with K-best algorithm and the coarse-granularity sorting strategy, computation complexity can be significantly reduced. Moreover, theoretical complexity

analysis of is presented, [1] providing the design parameters for early-pruned multi-K-best algorithm. The analysis also shows that the computation complexity of the proposed schemes is polynomial with the sphere degree.

[1]Acknowledgements are dedicated to Chien Ching Lin for his considerable contribution on the analysis for Fig.3.3, Fig.3.4, Figure 3.6, Figure 3.11, and Figure 3.12.



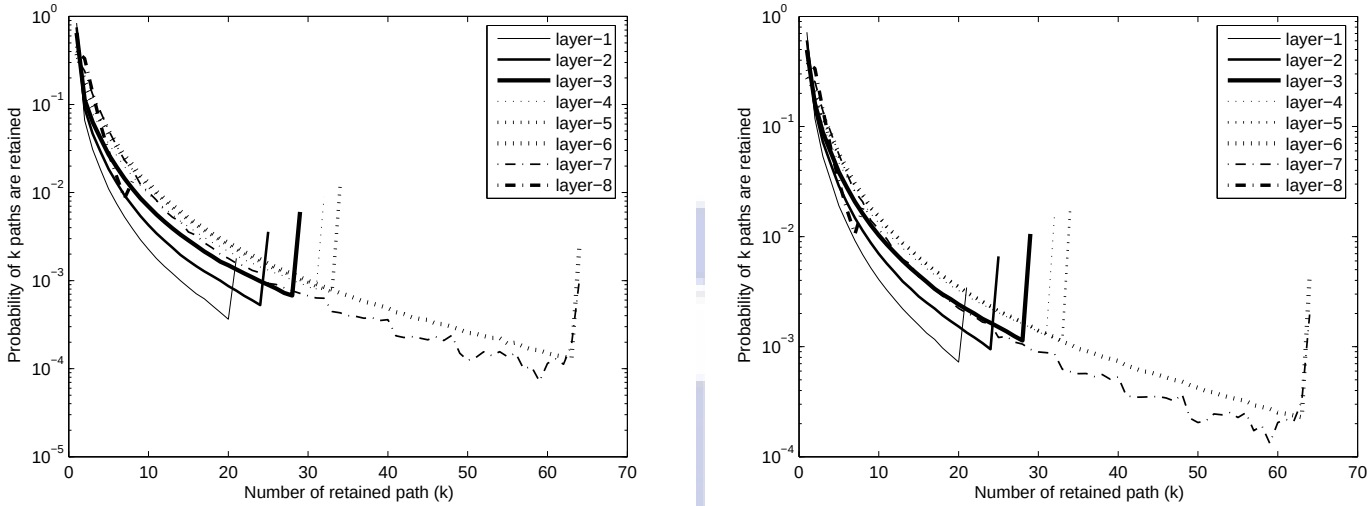
(a) SNR = 16dB

(b) SNR = 20dB

(c) SNR = 26dB

(d) SNR = 30dB

Figure 3.9: Simulated probability of retained paths for EP-64-best algorithm.



(a) SNR = 16dB (b) SNR = 20dB

(c) SNR = 26dB (d) SNR = 30dB

Figure 3.10: Simulated probability of retained paths for EP-multi-K-best algorithm.

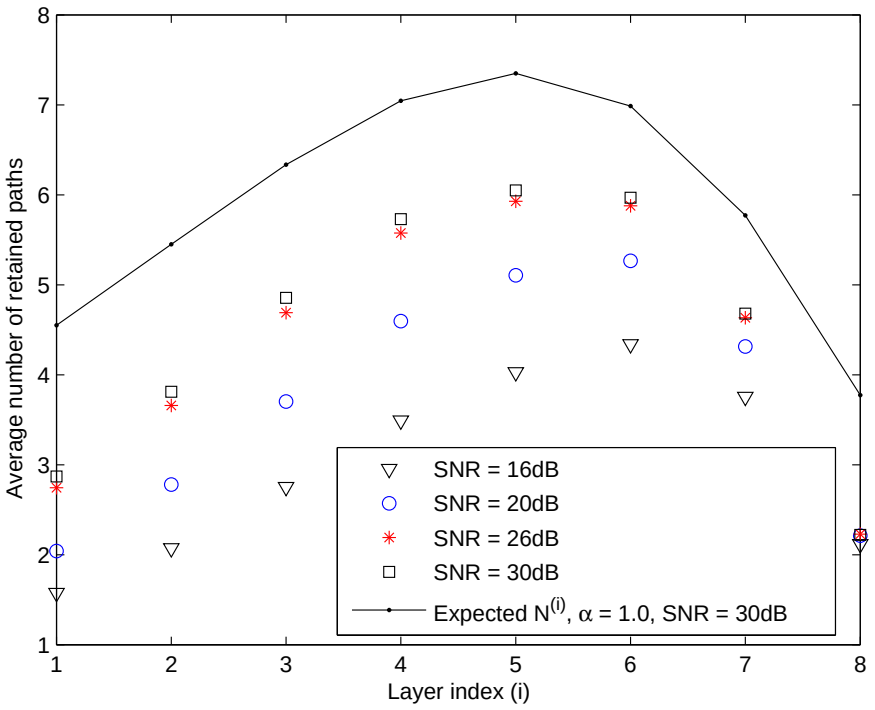


Figure 3.11: Average number of path retained at each layer for EP-64-best algorithm.

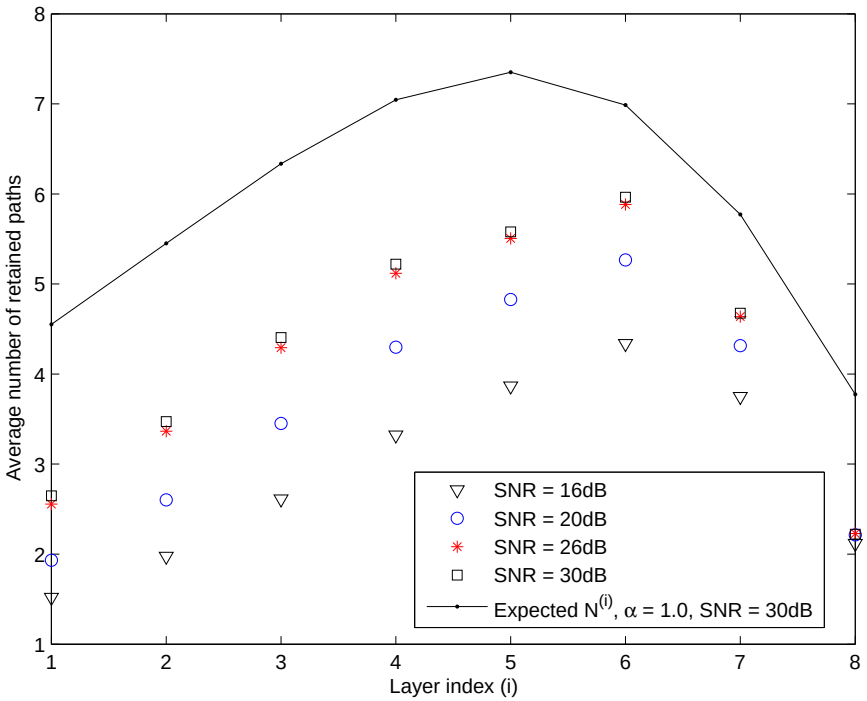


Figure 3.12: Average number of path retained at each layer for EP-multi-K-best algorithm.

Chapter 4

Low Density Parity Check Code Decoder

In 1963, Gallager [7] first introduced and proved low-density parity-check (LDPC) code as a powerful error control scheme. Until the advances in VLSI technology, LDPC codes were almost forgotten in the subsequent thirty years. Rediscovered by Mackay [8, 9] and then shown to be capacity-approaching [10–13], interests in LDPC codes eventually rose in the late 1990s. The simple arithmetic computations and implicit parallelism of the LDPC decoding algorithms facilitate low-complexity and high-speed hardware implementations. Now, many advanced communication systems such as digital television broadcasting (DVB-S2 [15], DMB-TH [16]), wireless local area network (IEEE802.11n [17]), wireless metropolitan network (IEEE802.16e [18]), and 10G BASE-T Eithernet (IEEE802.3an [19]), employ LDPC codes as the forward error correction (FEC) technique.

Being linear block codes, an LDPC code can be characterized by a sparse parity check matrix $\boldsymbol{H}$ which has only a small fraction of non-zero entries. The sparseness of $\boldsymbol{H}$ inherently reduces the computations in decoding. Moreover, $\boldsymbol{H}$ has a graphical representation [14, 73] where the rows and columns are associated to *check nodes* and *bit nodes*, respectively. The number of non-zero entries of each row or column is related to the degree of the corresponding check node or bit node. An LDPC code has the same check node degree and bit node degree is called a regular LDPC code. Otherwise, it will be referred to an irregular LDPC code.

Message-passing algorithm, also named *belief-propagation (BP) algorithm* [7, 9, 14], de-

codes LDPC codes by iteratively exchanging probabilistic information between check nodes and bit nodes. Moreover, the messages passed around are often represented by log-likelihood ratios (LLR) where the multiplications are transformed to additions, leading to reduction in computation complexity. However, some nonlinear operations are introduced. In this chapter, approximations for the nonlinear operation in decoding LDPC codes are discussed. A dynamic normalization technique will be introduced. Besides, analysis based on *order statistics* [31, 32] and *density evolution* [33] will be presented for deriving the normalization factors.

4.1 LDPC Decoding Algorithm

An N-bit LDPC code can be defined by an $M \times N$ parity check matrix $\boldsymbol{H} = [h_{mn}]$, where h_{mn} denotes the entry on the m-th row and n-th column of $\boldsymbol{H}$ for $1 \leq m \leq M$ and $1 \leq n \leq N$. Note that only binary LDPC codes will be considered hereafter. Same as every linear block code, each valid LDPC codeword $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ satisfies the parity check equations of $\mathbf{Hx} = \mathbf{0}$. Maximum likelihood (ML) decoding is equivalent to searching for the most likely codeword subject to $\mathbf{Hx} = \mathbf{0}$. However, exhaustive search is infeasible when codeword length N is large. Belief-propagation (BP) algorithm [7, 14] is one common approach to decode LDPC codes.

Tanner graph [73], which is also a bipartite graph [74], is one common graphical representation for the parity checks of an LDPC code. Figure 4.1 is an illustrative example of a 3×6 parity check matrix $\boldsymbol{H}$ and its corresponding Tanner graph. There are six *bit nodes*, BN_1, $BN_2, \ldots, BN_6$, representing the 6-bit codeword $\mathbf{x} = [x_1, x_2, \ldots x_6]^T$ and three *check nodes*, CN_1, CN_2, and CN_3, representing the three parity check equations of $\boldsymbol{H}$. Moreover, $M(n) = \{m : h_{mn} = 1\}$ is the set that check nodes connected to BN_n, and

$N(m) = \{n : h_{mn} = 1\}$ denotes the bit nodes connected to CN_m. The number of edges connected to a node is referred to the degree of the node. By definition, a regular LDPC code has equal check node degree and bit node degree, whereas the ones with different check node and bit node degrees are referred to irregular LDPC codes.

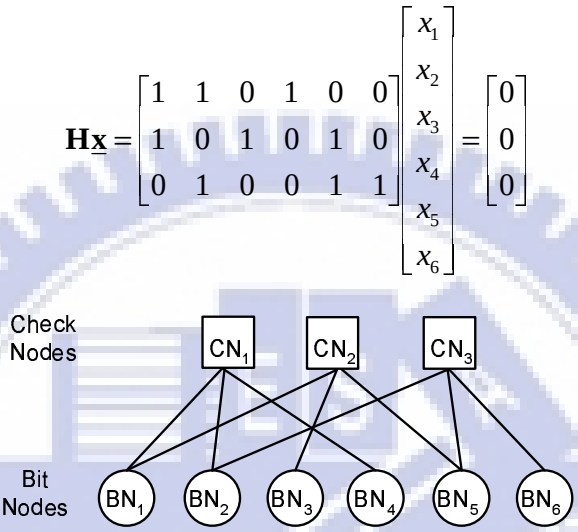


Figure 4.1: The parity check matrix and the corresponding Tanner graph

Let S_m be the event that the parity check equations of CN_m are satisfied. In each decoding iteration, the check node CN_m updates its outgoing message by the probability $P(S_m|x_{n'} = b)$, for all $n' \in N(m)$ and $b \in \{0, 1\}$. After the bit node BN_n receives all the messages from the check nodes in $M(n)$, the bit node updates its message according to the probability $P(x_n = b|S_{m'}, y_n)$, where $m' \in M(n)$ and y_n is the value received from the channel. Each bit node can accumulate more reliable information from the others by iteratively exchanging information between bit nodes and check nodes. The iterative decoding process proceeds until a valid codeword is found or the decoding iteration exceeds a predefined number. If the probabilistic messages are represented by log-likelihood ratios (LLR), Log-BP algorithm can be described as follows.

1. **Initialization** Under the assumption of equal priori, $P(x_n = 0) = P(x_n = 1) = 0.5$, the decoder calculates p_n, the intrinsic information of BN_n, by

$$p_n = \log \frac{P(y_n|x_n = 0)}{P(y_n|x_n = 1)}.$$

The message from BN_n to CN_m, denoted by q_{nm}, is initialized by $q_{nm} = p_n$, while the message from CN_m to BN_n, denoted by r_{mn}, is set to zero.

2. **Iterative Decoding**

a) *Bit Node Updating*

BN_n updates the message to CN_m by

$$q_{nm} = p_n + \sum_{m' \in \{M(n) \setminus m\}} r_{m'n}, \tag{4.1}$$

where the set $\{M(n) \setminus m\}$ contains all elements in $M(n)$ excluding m. Meanwhile BN_n decodes the n-th bit $\hat{x}_n$ by

$$\hat{x}_n = \begin{cases} 0, & \text{if } p_n + \sum_{m' \in M(n)} r_{m'n} \geq 0; \\ 1, & \text{otherwise.} \end{cases}$$

The iterative process terminates when a valid codeword $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N]^T$ is found, i.e. $\mathbf{H}\hat{\mathbf{x}} = \mathbf{0}$, otherwise the *Check Node Updating* continues. If the iteration number exceeds a predefined value, the decoder claims a decoding failure and terminates the decoding procedure.

b) *Check Node Updating*

CN_m updates r_{mn}, the message sent to BN_n, according to the messages received from

$\{N(m) \setminus n\}$ in which n is excluded:

$$r_{mn} = \prod_{n' \in \{N(m) \setminus n\}} \operatorname{sgn}(q_{n'm}) \times \Psi^{-1}\left(\sum_{n' \in \{N(m) \setminus n\}} \Psi(|q_{n'm}|) \right), \tag{4.2}$$

where

$$\Psi(a) = \Psi^{-1}(a) = \log \frac{1 + e^{-a}}{1 - e^{-a}}. \tag{4.3}$$

As it is shown in (4.2), the nonlinear function $\Psi(\cdot)$ is the most complicated operation in computing r_{mn}. Figure 4.2 illustrates the magnitude part of (4.2), where $q_1, q_2, \ldots, q_{d_c}$ represent the d_c check node input magnitudes. The nonlinear function $\Psi(\cdot)$ not only increases the implementation complexity, extensive quantization loss resulted from finite-precision representing $\Psi(\cdot)$ limits the error performance of the decoder. Thus, some approximation schemes had been proposed to facilitate circuit implementation.

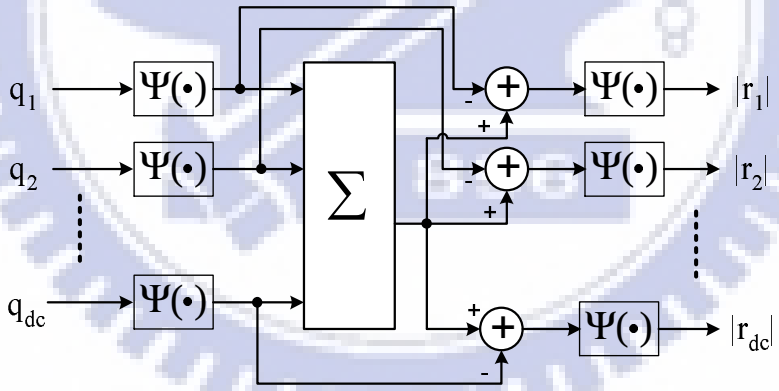


Figure 4.2: The architecture of the magnitude part of BP algorithm in (4.2)

Min-sum algorithm [20,21] discards the $(d_c - 2)$ smaller terms in the summation of (4.2) and approximates the check node updating by

$$r_{mn} \approx \prod_{n' \in \{N(m) \setminus n\}} \operatorname{sgn}(q_{n'm}) \min_{n' \in \{N(m) \setminus n\}} \{|q_{n'm}|\}. \tag{4.4}$$

However, there exists a performance gap between min-sum algorithm and Log-BP algorithm since min-sum algorithm always over-estimates the check node output magnitude. Several low-complexity approximations using a correction factor have then been introduced to compensate the performance loss [22–30, 75]. The compensation modifies the min-sum algorithm into the forms:

$$r_{mn} \approx \prod_{n' \in \{N(m) \backslash n\}} \operatorname{sgn}(q_{n'm}) \times \left(\min_{n' \in \{N(m) \backslash n\}} \{|q_{n'm}|\} - a \right) \tag{4.5}$$

or

$$r_{mn} \approx \prod_{n' \in \{N(m) \backslash n\}} \operatorname{sgn}(q_{n'm}) \times \left(\min_{n' \in \{N(m) \backslash n\}} \{|q_{n'm}|\} \times \beta \right), \tag{4.6}$$

where a and β are correction factors with $a > 0$ and $0 < \beta \leq 1$.

Recently, shuffled decoding [76, 77] has been proposed for better decoding convergence in the iterative process. The major difference between a standard BP decoder and a shuffled BP decoder lies in the message updating. The up-to-date messages computed at current iteration are used in shuffled BP algorithm, whereas the messages computed in previous iteration is used for standard BP algorithms.

Not only decoding convergence, the storage requirement in implementation can also be reduced by shuffled BP decoding. Two memory blocks are required for standard BP decoding; one is for the messages computed in the previous iteration and the other is for recording the messages computed at current iteration. But the two memory can be shared if applying shuffled BP decoding algortihm. Furthermore, potential improved decoding speed can be another benefit. Since the intra-iteration nd the inter-iteration no longer exist, about twice of the decoder throughput can be achieved by directly replacing standard BP with shuffled BP decoding.

However, shuffled BP decoding can be applied only in partially parallel or serial decoding scheduling. Otherwise, the messages are updated concurrently in a fully parallel decoder, and shuffled BP decoding will reduce to standard BP decoding.

4.2 Min-Sum algorithm with Dynamic Compensation

If a constant correction term in (4.5) or (4.6) is applied, they can be derived either empirically (by simulation) or theoretically (by analyzing statistics of the message distributions). For LDPC codes of long block length, density evolution [11, 23, 27–30, 33, 78] can be applied to determine the correction factor that is optimized for the channel parameters, noise variance for example. Except density evolution, averaging the difference between the min-sum approximation and the BP decoding is an alternatively intuitive approach. In [25], the normalization factors is determined by averaging the ratio of messages in min-sum and Log-BP algorithms; in [26], the correction factor is chosen such that the mean square error of approximation is minimized.

The derivations above only consider constant correction factors, however, constant factors are not always to provide sufficient performance improvement. Although [30] suggests two-dimensional normalization to reduce the performance gap between the constant normalized min-sum and Log-BP algorithms, each of the bit node and check node output messages are normalized by a constant still.

In fact, the normalization factor can be expressed as a function of the check node inputs, and more accurate approximation can be expected. In the following, we will present an analysis based on order statistics and density evolution for deriving the dynamic normalization factors.

4.2.1 Dynamic Normalization Factors

It can be easily verified that the magnitude part in (4.6) is equivalent to

$$|r_{mn}| = \begin{cases} m_1\beta_1, & \text{if } |q_{nm}| \neq m_1 \ ; \\ m_2\beta_2, & \text{otherwise.} \end{cases} \tag{4.7}$$

m_1 and m_2 are the minimum and second minimum among the check node input massage magnitudes. Note that each of m_1 and m_2 has a distinct normalization factor, β_1 and β_2. Let $q_1, q_2, \ldots, q_{d_c}$ represent the d_c magnitudes of a degree-d_c check node, Figure 4.3 illustrates computation of (4.7). Subsequently, let m_j be the j-th order statistic [31,32], i.e., $m_1 \leq m_2 \leq \cdots \leq m_{d_c}$. If the normalized min-sum algorithm (4.7) accurately represents Log-BP algorithm in (4.2), we must have

$$\begin{aligned} \Psi^{-1}\left(\sum_{n'\in\{N(m)\backslash n\}} \Psi(|q_{n'm}|)\right) &\approx \Psi^{-1}\left(\sum_{i=1}^{d_c} \Psi(m_i)\right) \\ &= m_1\beta_1 \end{aligned} \tag{4.8}$$

if $|q_{nm}| \neq m_1$, and

$$\begin{aligned} \Psi^{-1}\left(\sum_{n'\in\{N(m)\backslash n\}} \Psi(|q_{n'm}|)\right) &= \Psi^{-1}\left(\sum_{i=2}^{d_c} \Psi(m_i)\right) \\ &= m_2\beta_2 \end{aligned} \tag{4.9}$$

for $|q_{nm}| = m_1$. The normalization factors are defined by

$$\beta_1 \triangleq \frac{\Psi^{-1}(\sum_{j=1}^{d_c} \Psi(m_j))}{m_1}; \tag{4.10}$$

$$\beta_2 \triangleq \frac{\Psi^{-1}(\sum_{j=2}^{d_c} \Psi(m_j))}{m_2}. \tag{4.11}$$

That is, β_1 and β_2 are distinct functions of the check node inputs. Thus, the data-dependent normalization factors can provide a more accurate approximation. As Figure 4.4 shows, the function $\Psi(m)$ decays rapidly with m. As a result, for all $\Psi(m_{K+1}), \Psi(m_{K+2}), \ldots, \Psi(m_{d_c})$ that are relatively smaller than $\Psi(m_K)$, d_c-dimensional functions (4.10) and (4.11) can be simplified to K-dimensional functions as

$$\beta_1(m_1, m_2, \ldots, m_K) \approx \frac{\Psi^{-1}(\sum_{j=1}^{K} \Psi(m_j) + \sum_{j=K+1}^{d_c} E[\Psi(m_j)|m_K]}{m_1}; \tag{4.12}$$

$$\beta_2(m_2, m_3, \ldots, m_{K+1}) \approx \frac{\Psi^{-1}(\sum_{j=2}^{K+1} \Psi(m_j) + \sum_{j=K+2}^{d_c} E[\Psi(m_j)|m_{K+1}]}{m_2}. \tag{4.13}$$

All the $\Psi(m_j)$'s are approximated by the conditional expected values $E[\Psi(m_j)|m_K]$ for $j = K, K+1, \ldots, d_c$.

For LDPC codes of relatively large codeword length, $q_1, q_2, \ldots, q_{d_c}$ can be regarded i.i.d. with $f(m)$ and $F(m)$ as the pdf and cdf. Then $f_j(m)$, the pdf of j-th order statistic m_j, is [31, 32]

$$f_j(m) = \frac{d_c!}{(j-1)!(d_c-j)!}[F(m)]^{j-1} \times [1-F(m)]^{d_c-j} f(m), \tag{4.14}$$

for all $j = 1, 2, \ldots, d_c$. Consequently, $E[\Psi(m_j)|m_K]$ and $E[\Psi(m_j)|m_{K+1}]$ in (4.12) and (4.13)

can be computed by

$$E[\Psi(m_j)|m_K] = E[\Psi(m_j)|m_j \geq m_K] = \frac{\int_{m_K}^{\infty} \Psi(m) f_j(m) dm}{\int_{m_K}^{\infty} f_j(m) dm}, \tag{4.15}$$

and

$$E[\Psi(m_j)|m_{K+1}] = E[\Psi(m_j)|m_j \geq m_{K+1}] = \frac{\int_{m_{K+1}}^{\infty} \Psi(m) f_j(m) dm}{\int_{m_{K+1}}^{\infty} f_j(m) dm}. \tag{4.16}$$

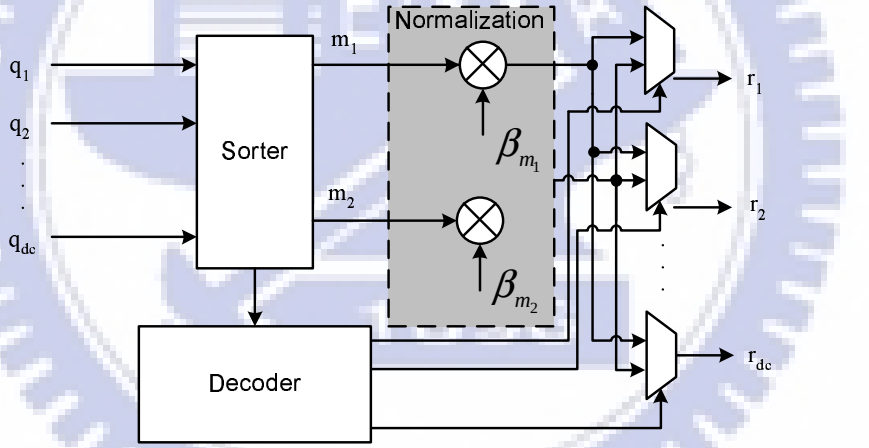


Figure 4.3: Realizing normalized min-sum algorithm of (4.6) by sorting.

4.2.2 Message Distribution under Iterative Decoding

The K-dimensional normalization factors defined in (4.12) and (4.13) can be computed by (4.15) and (4.16) when the degree and the input distributions are known. However, the check node input distributions vary with the decoding iteration under message-passing algorithm. Moreover, the distribution of first iteration is also determined according to the channel

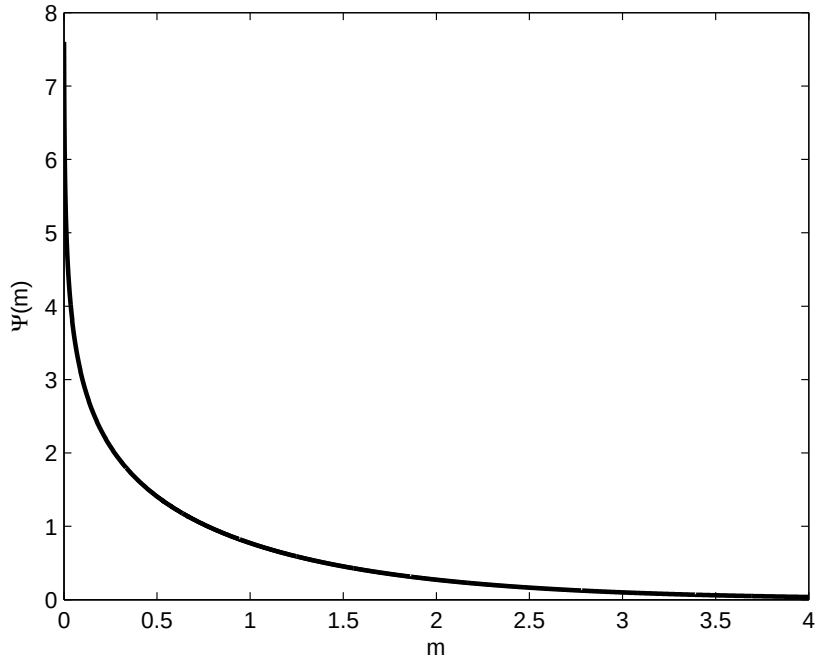


Figure 4.4: The $\Psi(m)$ function and $\Psi(m)$ decays rapidly as m increases.

statistics.

Density evolution is a technique to trace the message distribution under iterative decoding and can be applied here for analyzing the dynamic normalization factors. Because the sign and the magnitude of a check node can be updated separately, it is more convenient to represent the pdfs of the messages $q_{n'm}$ and $r_{m'n}$ in (4.1) and (4.2) by the *sign-magnitude* representation. That is, the message pdfs $f_Q(q)$ and $f_R(r)$ corresponding to a bit node and a check node will be represented by two-dimensional quantities $[S_Q,\ f_{|Q|}(q)]$ and $[S_R,\ f_{|R|}(r)]$, where the notation q and r stand for all $q_{n'm}$ and $r_{m'n}$. The terms S_Q and S_R represent the probability of q and r having positive signs, which are calculated by

$$S_Q = \int_0^\infty f_Q(q) d_q \tag{4.17}$$

and

$$S_R = \int_0^\infty f_R(r) d_r. \tag{4.18}$$

The second term $f_{|Q|}(q)$ and $f_{|R|}(r)$ are the pdfs of the magnitude of q and r, which can be derived by

$$f_{|Q|}(q) = f_Q(q) + f_Q(-q) \tag{4.19}$$

and

$$f_{|R|}(r) = f_R(r) + f_R(-r), \tag{4.20}$$

for $q \geq 0$ and $r \geq 0$.

It has been proved that the performance of an LDPC decoder is independent of the codeword as long as the symmetry conditions are satisfied [10]. Hence we assume an all-zero codeword $\mathbf{x} = \mathbf{0}$ is transmitted to reduce the computation complexity of the following analysis. Without loss of generality, randomly and equal prior data are transmitted in binary phase-shift keying (BPSK) signaling. Beside, a zero vector is assumed to be transmitted through an additive white Gaussian noise (AWGN) channel and corrupted by a noise vector $\mathbf{v}$, a sequence of independent Gaussian random variables with variance σ^2 and zero mean. Thus, the received signal $\mathbf{y} = \mathbf{1} + \mathbf{v}$ is also a sequence of independent Gaussian random variables with unity-mean and variance σ^2. Furthermore, the initial message of bit node BN_n becomes $p_n = \log \frac{P(y_n|x_n=0)}{P(y_n|x_n=1)} = \frac{2}{\sigma^2} y_n$, a Gaussian random variable with mean and variance equal to $\frac{2}{\sigma^2}$ and $\frac{4}{\sigma^2}$. With these assumptions, the distribution of messages and the normalization factors of the l-th decoding iteration can be acquired recursively through the following procedure.

Corollary 4.1. For two independent random variables Θ_1 and Θ_2 with pdfs $f_{\Theta_1}(\theta)_1$, $f_{\Theta_2}(\theta)_2$

and sign-magnitude pdf pairs

$$\bar{\Upsilon}_1 = [S_{\Theta_1},\ f_{|\Theta_1|}(\theta_1)]$$
$$\bar{\Upsilon}_2 = [S_{\Theta_2},\ f_{|\Theta_2|}(\theta_2)],$$

the pdf of $\Phi = \Theta_1 + \Theta_2$ is

$$\begin{aligned} f_\Phi(\phi) &= S_{\Theta_1} S_{\Theta_2} \int_{|\theta_1|+|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\theta \\ &+ (1 - S_{\Theta_1}) S_{\Theta_2} \int_{-|\theta_1|+|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\theta \\ &+ S_{\Theta_1}(1 - S_{\Theta_2}) \int_{|\theta_1|-|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\theta, \\ &\text{for } \phi < 0; \end{aligned} \tag{4.21}$$

$$\begin{aligned} f_\Phi(\phi) &= (1 - S_{\Theta_1})(1 - S_{\Theta_2}) \times \int_{-|\theta_1|-|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\theta \\ &+ (1 - S_{\Theta_1}) S_{\Theta_2} \int_{-|\theta_1|+|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) \theta \\ &+ S_{\Theta_1}(1 - S_{\Theta_2}) \int_{|\theta_1|-|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) \theta, \\ &\text{for } \phi \geq 0. \end{aligned} \tag{4.22}$$

Moreover, the sign-magnitude pdf pair

$$[S_\Phi,\ f_{|\Phi|}(\Phi)], \tag{4.23}$$

can be derived by

$$S_\Phi = \int_0^\infty f_\Phi(\phi) d_\phi \tag{4.24}$$

and

$$f_{|\Phi|}(\phi) = f_{\Phi}(\phi) + f_{\Phi}(-\phi) \tag{4.25}$$

for $\phi \geq 0$.

Corollary 4.2. Let $f_{\Theta_i}(\theta_i)$ and $\bar{\Upsilon}_i = [S_{\Theta_i},\ f_{|\Theta_i|}(\theta_i)]$, for $i = 1, 2, \ldots, N$, be the pdf and the equivalent sign-magnitude pdf pair of N independent random variables $\Theta_1, \Theta_2, \ldots, \Theta_N$. The sign-magnitude representation for the pdf of

$$\Phi = \sum_{i=1}^{N} \Theta_i \tag{4.26}$$

can be derived recursively by

$$\mathcal{A}\left(\cdots \mathcal{A}\left(\mathcal{A}\left(\bar{\Upsilon}_1, \bar{\Upsilon}_2\right), \bar{\Upsilon}_3\right), \cdots, \bar{\Upsilon}_N\right), \tag{4.27}$$

where

$$\bar{\Upsilon}_i = [S_{\Theta_i},\ f_{|\Theta_i|}(\theta_i)] \tag{4.28}$$

and $\mathcal{A}(\cdot)$ represent the process of deriving the sign-magnitude pdf pair of $\theta_i + \theta_j$ based on Corollary4.1.

Corollary 4.3. Let $\bar{\Upsilon}_i = [S_{\Theta_i},\ f_{|\Theta_i|}(\theta_i)]$ denote the sign-magnitude representation of the pdf of a random variable Θ_i, i.e $f_{\Theta_i}(\theta_i)$. Then, the corresponding sign-magnitude representation for the mixture of the pdfs

$$\sum_{i=1}^{N} \rho_i f_{\Theta_i}(\theta_i)$$

will be

$$\sum_{i=1}^{N} \rho_i \bar{\Upsilon}_i = \left[\sum_{i=1}^{N} \rho_i S_{\Theta_i}, \sum_{i=1}^{N} \rho_i f_{|\Theta_i|}(\theta_i)\right]. \tag{4.29}$$

Corollary 4.4. $M_1, M_2, \ldots, M_K$ are independent random variables and β is a K-dimensional function of $M_1, M_2, \ldots, M_K$. Let the pdfs and cdfs of M_j, $j = 1, 2, \ldots, K$, be denoted as $f_j(m)$ and $F_j(m)$. The pdf of the random variable $R = M_1\beta(M_1, M_2, \ldots, M_K)$ will be derived in the following.

Let $R = r, M_1 = m_1, M_2 = m_2, \ldots, M_K = m_K$ be one set of the solutions to

$$R = M_1\beta(M_1, M_2, \ldots, M_K). \tag{4.30}$$

Thus, for $m_2, m_3, \ldots, m_K$ are fixed, r is a function of m_1 only. Then, for all m_1 and the given $m_2, m_3, \ldots, m_K$, the pdf of R can be expressed by the pdf of M_1 as

$$\begin{aligned} f_R(r|m_2, m_3, \ldots, m_K) &= \sum_{m_1} f_{M_1}(m_1) \left|\frac{dr}{dm_1}\right|^{-1} \\ &= \sum_{m_1} f_{M_1}(m_1) \\ &\times \left|\beta(m_1, m_2, \ldots, m_K) + m_1\beta'(m_1, m_2, \ldots, m_K)\right|^{-1}, \end{aligned} \tag{4.31}$$

where

$$\beta'(m_1, m_2, \ldots, m_K) = \frac{d}{dm_1}\beta(m_1, m_2, \ldots, m_K). \tag{4.32}$$

Therefore, for all solutions to

$$r = m_1\beta(m_1, m_2, m_3, \ldots, m_K), \tag{4.33}$$

the pdf $f_R(r)$ will be

$$f_R(r) = \sum_{m_1} \sum_{m_2} \cdots \sum_{m_K} F'_{M_2}(m_2) F'_{M_3}(m_3) \cdots F'_{M_K}(m_K) \times f_{M_1}(m_1) \Big| \beta(m_1, m_2, \ldots, m_K) + m_1 \beta'(m_1, m_2, \ldots, m_K) \Big|^{-1}$$

Based on **Corollary 4.1** to **Corollary 4.4** and density evolution technique, the pdfs of the messages can be derived recursively as follows.

- **Step 1** [Output distribution of a bit node]:

  Let the random variable Q_i represent the input message of a bit node of degree-i, and $[S_{Q_i}^{(l)},\ f_{|Q_i|}^{(l)}]$ be the equivalent sign-magnitude represented pdf pair. By (4.1), the output distribution can be calculated from its input distribution $[S_R^{(l-1)},\ f_{|R|}^{(l-1)}]$, which is also the overall output distribution of the check nodes at the $(l-1)$-th iteration. Then, the sign-magnitude pdf pair of the output message can be derived according to **Corollary 4.2**.

- **Step 2** [Input distribution of the check nodes]:

  The pdf of the check node's input, denoted by $f_Q^{(l)}$, is a mixture of the pdfs $f_{Q_i}^{(l)}$ derived from Step 1, and

  $$f_Q^{(l)} = \sum_i \rho_i f_{Q_i}^{(l)} \tag{4.34}$$

  where ρ_i denotes the probability that the check node's inputs are sent from a bit node of degree i, and $\sum_i \rho_i = 1$. According to **Corollary 4.3**, the sign-magnitude representation of $f_Q^{(l)}$ will be

  $$[S_Q^{(l)},\ f_{|Q|}^{(l)}] = \left[\sum_i \rho_i S_{Q_i}^{(l)},\ \sum_i \rho_i f_{|Q_i|}^{(l)} \right]. \tag{4.35}$$

- **Step 3** [Output distribution of a check node]:

  The output distribution of a check node will be calculated after its input distribution $[S_Q^{(l)},\ f_{|Q|}^{(l)}]$ is derived at Step 2. For a check node of degree i, the sign of the check node's output is determined according to the sign operation in (4.2), and all the inputs are assumed to be i.i.d. random variables, the probability $S_{R_i}^{(l)}$ that the output sign is positive will be

$$\begin{aligned} S_{R_i}^{(l)} &= \sum_{j:even} \binom{i}{j} (1-S_Q^{(l)})^j (S_Q^{(l)})^{i-j} \\ &= \frac{1}{2}[(1-S_Q^{(l)}+S_Q^{(l)})^i + (1-S_Q^{(l)}-S_Q^{(l)})^i] \\ &= \frac{1}{2}[1+(1-2S_Q^{(l)})^{i-1}]. \end{aligned} \tag{4.36}$$

  According to (4.7), the check node has only two output magnitudes

$$R_{i1} = M_1 \beta_1(M_1, M_2, \ldots, M_K) \tag{4.37}$$

  and

$$R_{i2} = M_2 \beta_2(M_2, M_3, \ldots, M_{K+1}). \tag{4.38}$$

  Then the pdfs of R_{i1} and R_{i2}, denoted by $f_{R_{i1}}(r)$ and $f_{R_{i2}}(r)$, will be expressed by the

pdfs $f_{M_1}^{(l)}(m)$, $f_{M_2}^{(l)}(m)$; and the cdfs $F_{M_j}^{(l)}(m)$ for $j = 2, 3, \ldots, K+1$. That is,

$$f_{R_{i1}}^{(l)}(r) = \sum_{x_1, x_2, \ldots, x_K} F'_{M_2}(x_2) F'_{M_3}(x_3) \cdots F'_{M_K}(x_K) f_{M_1}^{(l)}(x_1) \times \left| \beta_1^{(l)}(x_1, x_2, \ldots, x_K) + x_1 \frac{d}{dx_1} \beta_1^{(l)}(x_1, x_2, \ldots, x_K) \right|^{-1} \tag{4.39}$$

and

$$F'_{M_j}(x_j) = \frac{d}{dm} F_{M_j}^{(l)}(m) \Big|_{m=x_j}, j = 2, 3, \ldots, K \tag{4.40}$$

for all $x_1, x_2, \ldots, x_K$ such that

$$x_1 \beta_1(x_1, x_2, \ldots, x_K) = r;$$

and

$$f_{R_{i2}}^{(l)}(r) = \sum_{x_2, x_3, \ldots, x_{K+1}} F'_{M_3}(x_3) F'_{M_4}(x_4) \cdots \times F'_{M_{K+1}}(x_{K+1}) f_{M_2}^{(l)}(x_2) \times \left| \beta_2^{(l)}(x_2, \ldots, x_{K+1}) + x_2 \frac{d}{dx_2} \beta_2^{(l)}(x_2, \ldots, x_{K+1}) \right|^{-1} \tag{4.41}$$

$$F'_{M_j}(x_j) = \frac{d}{dm} F_{M_j}^{(l)}(m) \Big|_{m=x_j}, j = 3, 4, \ldots, K+1 \tag{4.42}$$

for all $x_2, x_3, \ldots, x_{K+1}$ such that

$$x_2\beta_2(x_2, x_3, \ldots, x_{K+1}) = r. \tag{4.43}$$

Note that $f^{(l)}_{R_{i1}}(r)$ and $f^{(l)}_{R_{i2}}(r)$ can be derived by **Corollary 4.4**. Furthermore, we can see from (4.7) that only one of the output messages will have magnitude R_{i2}, and the others will have magnitude R_{i1}. The check node output magnitude $|R_i|$ will have the distribution

$$f^{(l)}_{|R_i|}(r) = \frac{i-1}{i} f^{(l)}_{R_{i1}}(r) + \frac{1}{i} f^{(l)}_{R_{i2}}(r). \tag{4.44}$$

- **Step 4** [Input distribution of the bit nodes]:

  The input distribution of a bit node can be calculated by a mixture of the pdfs for check nodes of different degrees. That is $f^{(l)}_R = \sum_i \lambda_i f^{(l)}_{R_i}$ where $f^{(l)}_{R_i}$ is the output distribution of a check node of degree i, λ_i denotes the probability of the messages coming from a check node of degree i, and $\sum_i \lambda_i = 1$. Based on Appendix B, the input distribution of a bit node can be calculated by $[S^{(l)}_R, f^{(l)}_{|R|}] = [\sum_i \lambda_i S^{(l)}_{R_i}, \sum_i \lambda_i f^{(l)}_{|R_i|}]$, and can be used for the analysis of the $(l+1)$-th iteration.

Subsequently, repeat from Step 1 to Step 4, the distribution of the messages and the normalization factors of each decoding iteration can be derived.

As the channel condition is given, the normalization factors of a specific LDPC code can be analyzed by (4.12)-(4.16) and the 4-step procedure as mentioned above. Figure 4.5 and Figure 4.6 illustrate the normalization factors of the 64,800-bit, $R = \frac{3}{5}$ LDPC code specified in DVB-S2 [15] BPSK signaling under AWGN channel. Figure 4.5 illustrates β for $K = 1$ at different decoding iteration and SNR while Figure 4.6 plots β for $K = 2$ at the first iteration and SNR = 2.2 dB. Note that large m_1 or m_2 will require larger normalization

factors. Furthermore, it can also be observed in Figure 4.5 that β increases with the iteration number and the channel SNR.

4.3 Implementation of Dynamic Normalization

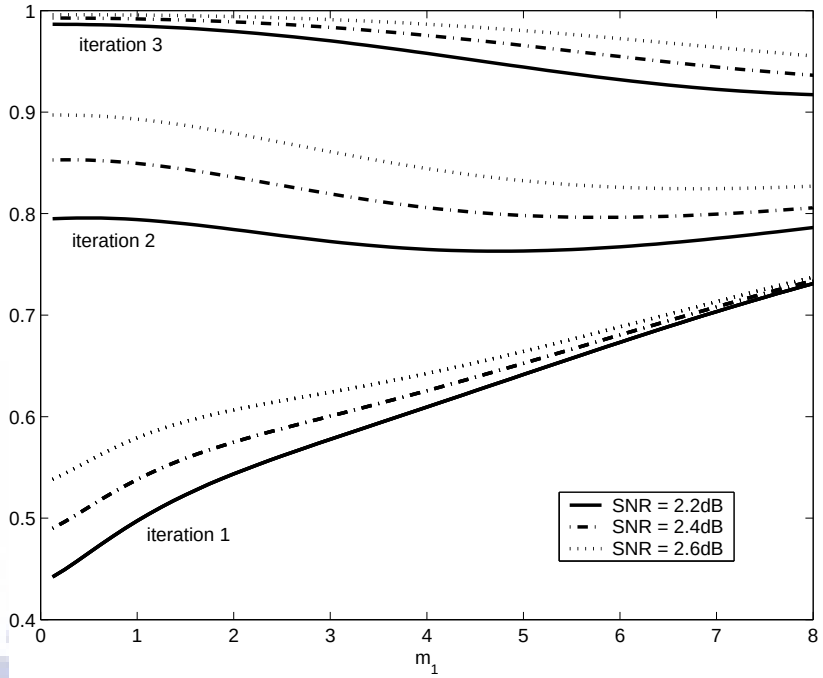
So far, we have presented a means to estimate the normalization factors for each decoding iteration. Although the distributions are analyzed at design time, the K-dimensional forms derived from (4.12) to (4.16) are still complicated. Considering the implementation complexity, further simplification on the normalization factors are required. Applying different normalization factors at different decoding iteration will be costly in hardware implementation. Averaging the normalization factors over several iteration is a straightforward approach to realize the dynamic normalization, by which the normalization factors become iteration-irrelative. When given the channel SNR, the normalization factors become functions of the $(K+1)$ smallest check node input magnitudes $m_1, m_2, \ldots, m_{K+1}$

$$\beta_{m_1} = \frac{\sum_l \beta_1^{(l)}(m_1, m_2, \ldots, m_K) P^{(l)}(m_1, m_2, \ldots, m_K)}{\sum_l P^{(l)}(m_1, m_2, \ldots, m_K)} \tag{4.45}$$

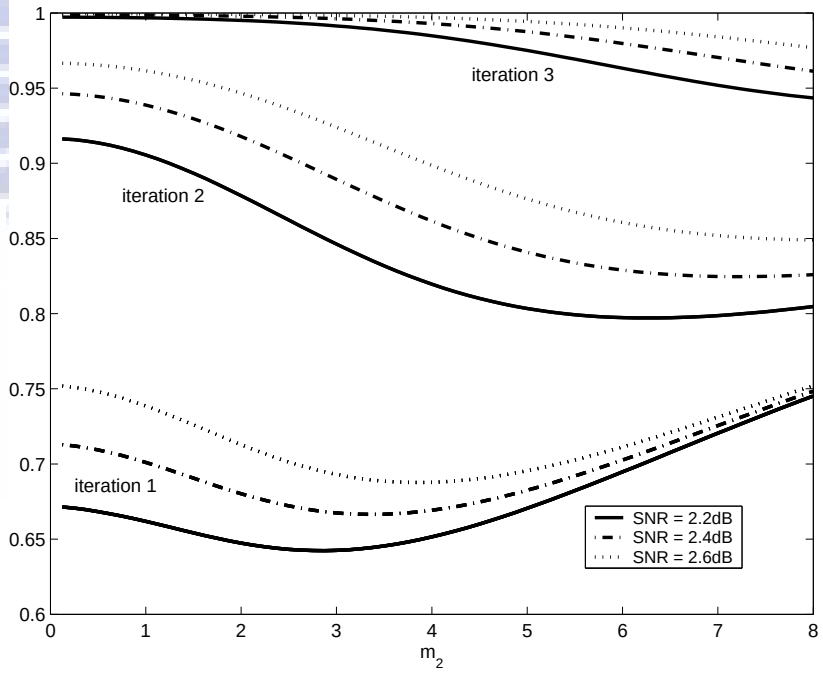
and

$$\beta_{m_2} = \frac{\sum_l \beta_2^{(l)}(m_2, m_3, \ldots, m_{K+1}) P^{(l)}(m_2, m_3, \ldots, m_{K+1})}{\sum_l P^{(l)}(m_2, m_3, \ldots, m_{K+1})} \tag{4.46}$$

where $\beta_1^{(l)}(m_1, m_2, \ldots, m_K)$ and $\beta_2^{(l)}(m_2, m_3, \ldots, m_{K+1})$ are the normalization factors for the l-th decoding iteration; $P^{(l)}(m_1, m_2, \ldots, m_K)$ and $P^{(l)}(m_2, m_3, \ldots, m_{K+1})$ are the probabilities of the check node having its $(K+1)$ smallest input magnitudes equaling to $m_1, m_2, \ldots, m_{K+1}$ at the l-th decoding iteration. Figure 4.7 illustrates the averages of (4.45) and (4.46) corresponding to Figure 4.5. Based on (4.45) and (4.46), three normalization approaches with different complexities will be presented.

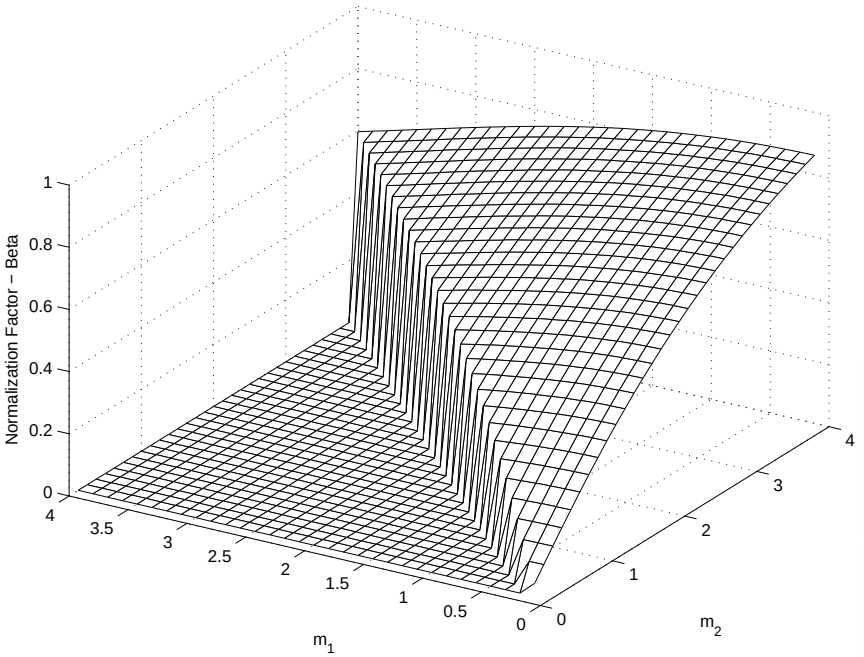


(a) $\beta_1(m_1)$

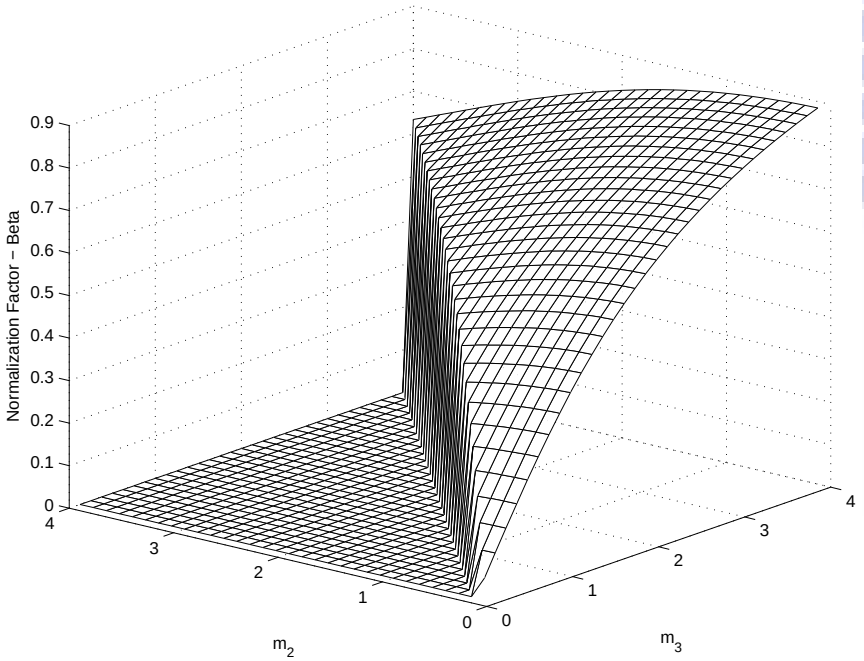


(b) $\beta_2(m_2)$

Figure 4.5: 1-D ($K = 1$) normalization factors $\beta_1(m_1)$ and $\beta_2(m_2)$ of the rate $\frac{3}{5}$, 64,800-bit LDPC code specified in DVB-S2.

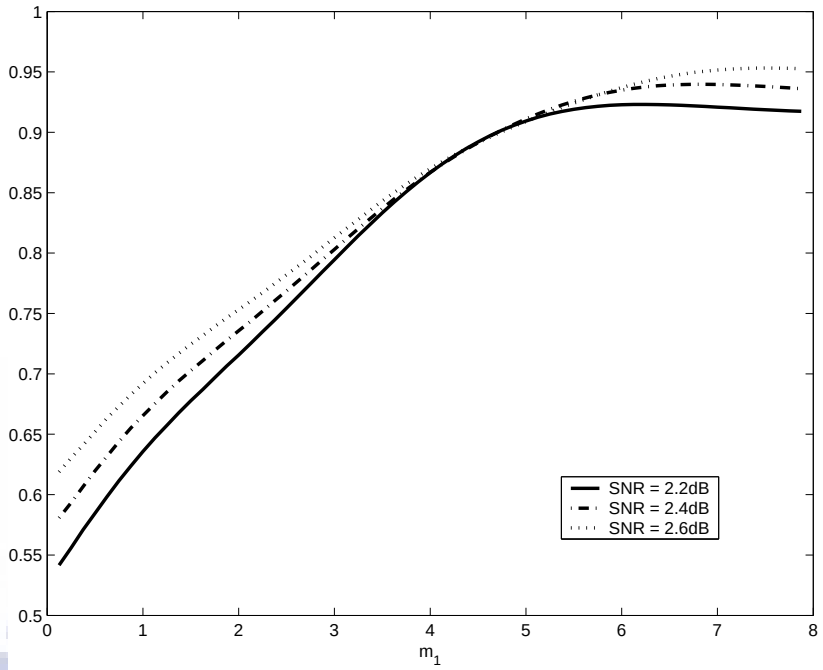


(a) $\beta_1^{(1)}(m_1, m_2)$

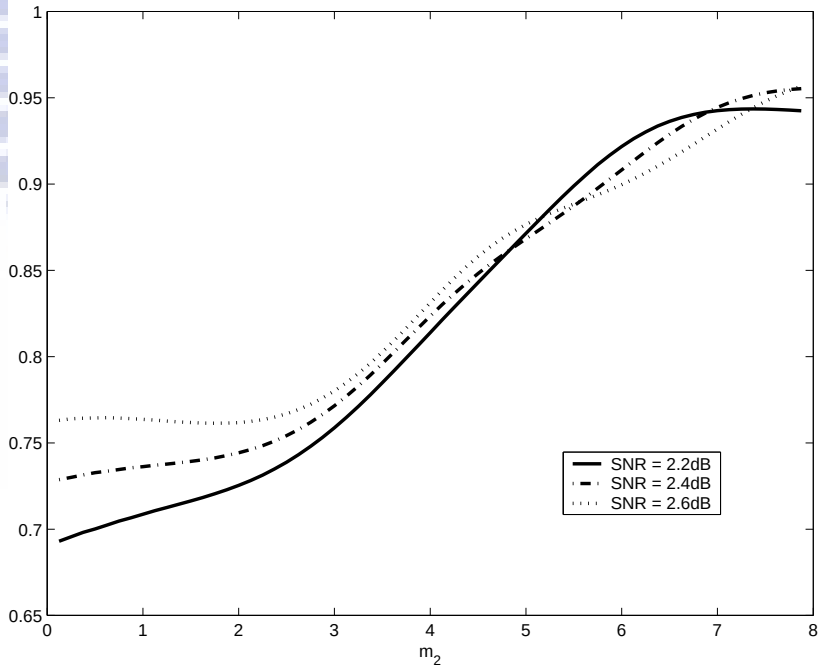


(b) $\beta_2^{(1)}(m_2, m_3)$

Figure 4.6: 2-D ($K = 2$) normalization factors $\beta_1(m_1, m_2)$ and $\beta_2(m_2, m_3)$ at the first decoding iteration for the rate $\frac{3}{5}$, 64,800-bit LDPC code specified in DVB-S2.



(a) Average of $\beta_1^{(l)}(m_1)$



(b) Average of $\beta_2^{(l)}(m_2)$

Figure 4.7: The averaged normalization factors in Figure 4.5.

4.3.1 Direct mapping approach

As Figure 4.8(a) shows, the normalization is implemented by two look-up tables (LUT) where $m_1, m_2, \ldots, m_{K+1}$ are directly mapped onto $m_1\beta_{m_1}$ and $m_2\beta_{m_2}$. This approach provides a straightforward and highly precise approximation of the nonlinear function. However, there exists overhead of storage requirement for the look-up tables.

4.3.2 Adaptive-β approach

This scheme confines the choice of β_{m_1} and β_{m_2} to N_R candidates, which are denoted as $\beta_{m_{1j}}$ and $\beta_{m_{2j}}$ for $j = 1, 2, \ldots, N_R$. Moreover, let Γ_1 and Γ_2 denote the range of the check node input magnitudes, which are also partitioned into N_R parts where $\Gamma_1 = \bigcap_{j=1}^{N_R} \Gamma_{1j}$ and $\Gamma_2 = \bigcap_{j=1}^{N_R} \Gamma_{2j}$. For all $[m_1, m_2, \ldots, m_K] \in \Gamma_{1j}$, and $[m_2, m_3, \ldots, m_{K+1}] \in \Gamma_{2j}$, the corresponding β_{m_1} and β_{m_2} will be assigned as β_{1j} and β_{2j} which minimize the average scaling error

$$\epsilon_s = \sum_{j=1}^{N_R} \left| 1 - \left(\overline{\frac{d_c - 1}{d_c} \frac{\beta_{m_{1j}}}{\beta_{m_1}} + \frac{1}{d_c} \frac{\beta_{m_{2j}}}{\beta_{m_2}}} \right) \right|, \tag{4.47}$$

where $\overline{x}$ denotes the average of x. In fact, it will be shown by our simulation results that $K = 1$ can provide a quite accurate approximation for (4.2). Let us define *single-β approach* for $K = 1, N_R = 1$ and *double-β approach* $K = 1, N_R = 2$. For double-β approach, the normalization factors β_{m_1} and β_{m_2} can be determined by

$$\beta_{m_1} = \begin{cases} \beta_{11}, & \text{if } m_1 \le T_1, \\ \beta_{12}, & \text{otherwise;} \end{cases} \tag{4.48}$$

$$\beta_{m_2} = \begin{cases} \beta_{21}, & \text{if } m_2 \le T_2, \\ \beta_{22}, & \text{otherwise,} \end{cases} \tag{4.49}$$

where T_1 and T_2 can be derived by uniformly partitioning the input range and adjusting empirically after the normalization factors are determined.

4.3.3 Annealing approach

Sometimes the min-sum algorithm could be compensated incorrectly due to the finite precision and limited candidates of normalization factors. On one hand, the normalization factors in (4.45) and (4.46) are averaged to the iteration number. However, the normalization factors tend to increase with iteration, the check node outputs may be over-normalized and the messages are equivalently scaled by a smaller factor. On the other hand, min-sum algorithm always over-estimates the check node updating; the check node output is equivalent to scaling by a factor that is greater than 1. To prevent error accumulating with decoding iteration, normalization may not be necessarily required every iteration. That is, normalization can be applied intermittently. For example, given an integer L and the iteration number l, normalization is applied only when $(l \bmod L) \neq L-1$. It is equivalent to scaling the correct check node outputs by another factor $\gamma_s, \gamma_s \geq 1$, when $(l \bmod L) = L-1$. For a check node of degree d_c, γ_s can be estimated to be

$$\gamma_s = \frac{d_c - 1}{d_c} \times \frac{1}{N_R} \sum_{j=1}^{N_R} \frac{1.0}{\beta_{m_{1j}}} + \frac{1}{d_c} \times \frac{1}{N_R} \sum_{j=1}^{N_R} \frac{1.0}{\beta_{m_{2j}}}, \tag{4.50}$$

where N_R is the number of available β's.

Besides, this annealing approach equivalently provides more choices of β in finite precision representation; we can derive other normalization factors by properly defining r and L when β_{m_1} and β_{m_2} are given. That is, the effect of scaling by γ_s should be balanced by the following $L-1$ iterations. Therefore the normalization factors at the $L-1$ iterations are equivalent

to $\tilde{\beta}_{m_1}$ and $\tilde{\beta}_{m_2}$, where

$$\tilde{\beta}_{m_1} = \beta_{m_1} \times \gamma_s^{-\frac{1}{L-1}} \tag{4.51}$$

and

$$\tilde{\beta}_{m_2} = \beta_{m_2} \times \gamma_s^{-\frac{1}{L-1}}, \tag{4.52}$$

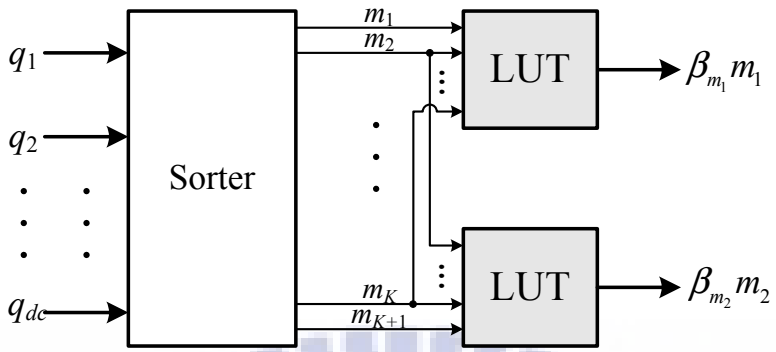
for all $L > 1$. When $L = 1$, $\gamma_s = 1$, then $\tilde{\beta}_{m_1} = \beta_{m_1}$ and $\tilde{\beta}_{m_2} = \beta_{m_2}$. Accordingly, more choices of β are available by varying L when β is restricted to finite number of candidates. Thus, a finer resolution of β can be realized without increasing the message bit-widths. Moreover, the annealing normalization reduces computation and facilitates a more power-efficient implementation. Figure 4.8(c) illustrates this annealing approach where the controller decides if the dynamic normalization should be applied according to the current iteration number l.

The following example demonstrates β_{m_1} and β_{m_2} derivation for different realization approaches. To further reduce the implementation complexity, the values β_{m_1} and β_{m_2} are restricted to $\sum_i 2^i$ such that the normalization circuits can be implemented by few shifters and adders.

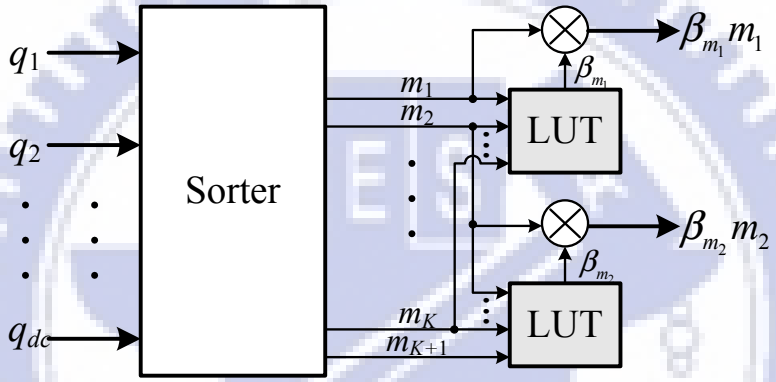
Example: The R-$\frac{3}{5}$, 64,800-bit LDPC code in DVB-S2 [15]:

1. Parameters for analyzing the normalization factors:

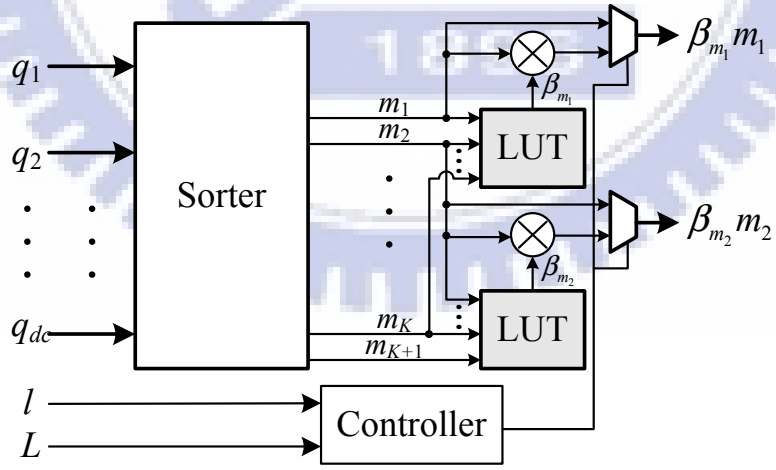
   There are $64800 \times \frac{2}{5} = 25920$ check nodes. Only one of the check node has degree 10 and the rest 25919 check nodes have degree 11. Let λ_i denote the probability that a messages coming from a check node of degree i. Then $\lambda_{11} \approx 1.0$ and $\lambda_{10} \approx 0$. Moreover, the probability of a bit node connecting to i check nodes is represented by $B_i = \frac{N_{B_i}}{N}$, where N_{B_i} is the number of bit node of degree i, N is the total number of bit nodes,



(a) Direct-mapping approach



(b) Adaptive-β approach



(c) Annealing approach for $K = 1$

Figure 4.8: Architectures of different realization of dynamic normalization

and $\sum_i B_i = 1$. Hence, the probability ρ_i defined in section III can be calculated by

$$\rho_i = \frac{NiB_i}{N\sum_i iB_i} = \frac{iB_i}{\sum_i iB_i}.$$

In this example, $N = 64800$ and $i = \{12, 3, 2\}$. Therefore $N_{B_{12}} = 12,960$, $N_{B_3} = 25,920$, and $N_{B_2} = 25,920$, leading to the following results: $B_{12} = 0.2$, $B_3 = 0.4$, $B_2 = 0.4$, and $\rho_{12} = 0.545$, $\rho_3 = 0.273$, $\rho_2 = 0.182$. Therefore, the normalization factors can be derived for different iterations based on the analysis in section III, and averaged to the decoding iteration according to (4.45) and (4.46). Moreover, we only consider the case $K = 1$ in this example.

2. Determine the normalization factors:

   The normalization factors will be restricted in the set $\{\frac{1}{8}, \frac{2}{8}, \ldots, 1\}$ for simple implementation. Besides, we only consider finite precision message representation that represents the maximum magnitude by 4.0.

   (a) Single-β approach: $\beta_{m_1} = 0.625$ and $\beta_{m_2} = 0.875$.

   (b) Double-β approach: The input is uniformly divided into two regions. Thus the threshold $T_1 = 2.0$ and $T_2 = 2.0$. Then the normalization factors that minimize (4.47) will be $\beta_{11} = 0.5$, $\beta_{12} = 0.75$, $\beta_{21} = 0.75$, $\beta_{22} = 1.0$.

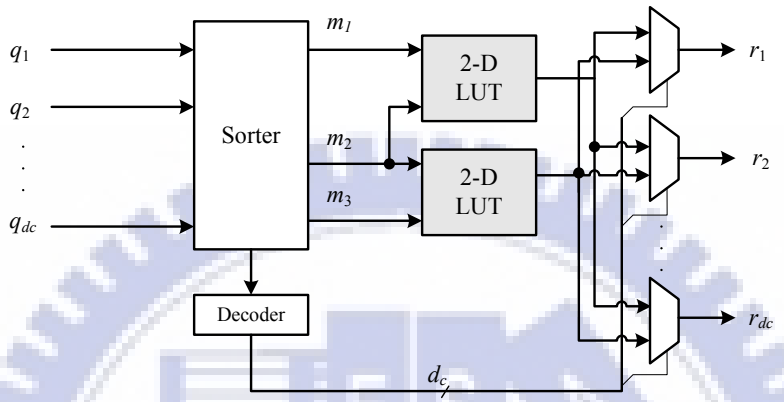
   (c) Annealing, single-β approach: $\gamma_s = 1.558$ for $L = 3$. By (4.51) and (4.52), $\tilde{\beta}_{m_1}$ and $\tilde{\beta}_{m_2}$ can then be determined to be $0.625\times(1.558)^{-\frac{1}{2}} = 0.501$ and $0.75\times(1.558)^{-\frac{1}{2}} = 0.701$, which will be modified into 0.5 and 0.75.

   (d) Annealing, double-β approach: $\gamma_s = 1.621$ for $L = 3$. Similarly, $\tilde{\beta}_{11}$, $\tilde{\beta}_{12}$, $\tilde{\beta}_{21}$, and $\tilde{\beta}_{22}$ will be 0.375, 0.625, 0.625, and 0.75 respectively.

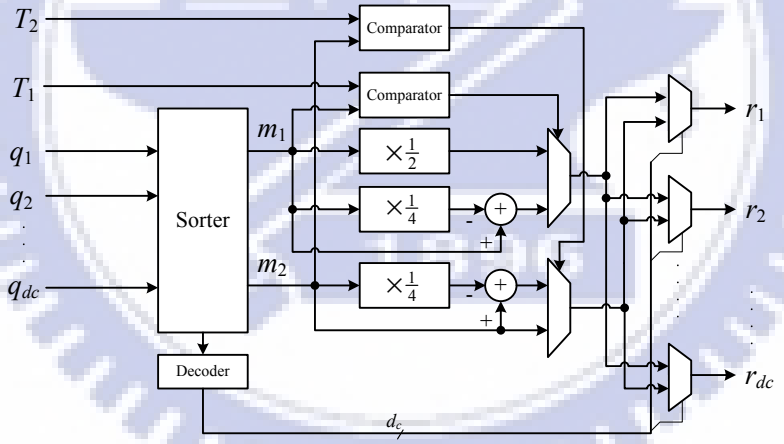
For the Annealing approach with single-β and $L = 3$, γ_s is 1.558 according to (4.50) where $\beta_{m_1} = 0.625$, $\beta_{m_2} = 0.875$ and $N_R = 1$. With (4.51) and (4.52), $\tilde{\beta}_{m_1}$ and $\tilde{\beta}_{m_2}$ can then be determined to be $0.625 \times (1.558)^{-\frac{1}{2}} = 0.501$ and $0.75 \times (1.558)^{-\frac{1}{2}} = 0.701$ and modified into 0.5 and 0.75, the closest candidates in Ω. Furthermore, the γ_s of the Annealing double-β approach is 1.621 according to (4.50), and the normalization factors, $\tilde{\beta}_{11}$, $\tilde{\beta}_{12}$, $\tilde{\beta}_{21}$, and $\tilde{\beta}_{22}$ will be 0.375, 0.625, 0.625, and 0.75 respectively.

Figure 4.9 illustrates two implementation approach for this example, the direct-mapping approach and the double-β normalization. The normalization scheme in Figure 4.9(a) is realized by a 2-dimensional (2-D) look-up-table (LUT), whereas the constant multiplications in Figure 4.9(b), $\times\frac{1}{2}$ and $\times\frac{1}{4}$, are performed by shifters.

In terms of area and timing, Fig.4.10 compares the circuit overheads in Figure 4.9(a) and Figure 4.9(b) to that of min-sum algorithm. The check node unit that has degree 11 and 5-bit messages is synthesized with the 0.13-μm cell library in either area critical or timing critical conditions. The gray portions in Figure 4.10 also present the additional gate count and timing contributed by the normalization circuit. Both figures show that the 2-D LUT direct-mapping normalization occupies about 50% of the gate count and 30% of the critical path delay due to the large LUT growing in quadratic with the bit-width of the messages. However, the double-β approach requires only additional shifters and adders, leading to 5% area (with 68 and 107 additional gates for each constraint) and 17% delay increases. It will be shown in next section that similar error performance can be achieved by these two schemes, however.



(a) The 2-D LUT direct-mapping approach for $K = 2$



(b) Adaptive-β approach for $K = 1$, $N_R = 2$

Figure 4.9: Architectures of the direct-mapping and the double-β approach for rate $\frac{3}{5}$, 64800-bit LDPC code in DVB-S2

4.4 Simulation Results

In the following, simulations based on 64,800-bit LDPC codes defined in DVB-S2 [15] are presented. More than 3000 frames of LDPC codes, which equals to $3000 \times 64800 \times Rate = 1.944 \times Rate \times 10^8$ bits, were simulated for each point. Moreover, belief-propagation algorithm with floating-point messages, abbreviated to BP-FP, is simulated as the baseline performance. Several aforementioned normalization approaches are compared. In the following, the adaptive-β approach with $K = 1, N_R = 1$ will be referred to *single-β* approach; the adaptive-β approach with $K = 1, N_R = 2$ will be referred as double-β approach; normalization by a constant will be referred to *fixed-β approach*.

The simulation channel is modeled as AWGN, and the randomly generated binary data is modulated by QPSK signaling before transmission, where the LDPC decoder can be initialized by the same method of BPSK. The maximum decoding iteration number is limited to 50. Except BP-FP, all the messages for different normalization approaches are represented by finite-precision; the bit-width of all messages are quantized to 6 bits. Considering low-complexity implementation, the normalization factors are restricted in the set $\{\frac{1}{8}, \frac{2}{8}, \ldots 1\}$ such that only few shifters and adders will be required.

4.4.1 Comparison of BP-FP and Min-Sum Algorithm

Table 4.1 compares the minimum working SNRs of BP-FP and min-sum algorithms, defined by the minimum SNR for bit error rate (BER) below 10^{-5}. Note that the signal to noise power ratio, SNR, is defined as

$$SNR = \frac{E_b}{N_0} + 10\log_{10}(2M_c \times Rate), \tag{4.53}$$

Table 4.1: The minimum working SNR of BP-FP and min-sum algorithm

Rate	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{5}{6}$	$\frac{8}{9}$	$\frac{9}{10}$
BP-FP (dB)	-2.55	-1.35	-0.45	0.9	2.15	3.05	3.95	4.6	5.1	6.15	6.35
Min-Sum (dB)	-2.05	-0.6	0.55	1.7	3.15	3.3	4.25	4.9	5.35	6.35	6.55
$\Delta_{MS-BP}(dB)$	0.5	0.75	1.0	0.8	1.0	0.25	0.3	0.3	0.25	0.2	0.2

where $2M_c$ bits are mapped to one complex symbol. The finite precision formate (a, b) means that $(a + b + 1)$ bits represent one message; where a bits are for the integer part and b bits are for the fractional part, and the one extra bit is for the sign of the message. Different combinations for (a, b) for a+b+1 = 6 has been simulated and the $(3, 2)$ format will contribute to the lowest error rate for min-sum algorithm for all rates. The term Δ_{MS-BP} is the SNR difference between the min-sum and BP-FP algorithms. According to Table 4.1, Δ_{MS-BP} is kept within 0.3dB for $R \geq \frac{2}{3}$ since the codes work in better channel conditions such that min-sum algorithm yields a good approximation. However, more accurate approximation is necessary to improve the performance when $R < \frac{2}{3}$. The proposed dynamic normalization will effectively reduce the performance loss caused by min-sum algorithm for those codes working at low SNR environments.

4.4.2 Comparison of Dynamic Normalization Approaches

As it is shown in Table 4.1, $R = \frac{2}{5}$ and $R = \frac{3}{5}$ correspond to the largest SNR loss. Therefore a discussion focused on the $R = \frac{3}{5}$ LDPC code will be presented since there is larger room for improvement. The resulted BER versus SNR for different normalization approaches are compared in Figure 4.11. All the corresponding parameters resulting in the best working SNR for different approaches are listed in Table 4.2 and Table 4.3. Note that the 2-D LUT

direct-mapping approach outperforms all the other normalization schemes, but has a great storage overhead. The double-β approach in Fig.4.11 has a comparable performance while requiring few additional logics for normalization.

In Figure 4.12, the limited maximum decoding iteration for different normalization approaches are compared. When the iteration number exceeds this maximum value, the iterative decoding terminates whether the codeword is decoded correctly or not. The proposed double-β normalization outperforms the fixed-β approach while the former requires maximum 20 decoding iterations and the latter requires maximum 50 decoding iterations to achieve BER $= 10^{-5}$ at similar SNR. Moreover, comparing the double-β normalization with min-sum algorithm, the former requires maximum 12 iterations and the later requires maximum 50 iterations to achieve BER $= 10^{-5}$ under the same SNR condition. Figure 4.12 shows that when the decoding complexity and speed are both critical, the proposed dynamic normalization improves the decoding speed of fixed-β and min-sum algorithm by about 60% and 76%, respectively.

In Table 4.4, the performance of several normalization schemes are compared for all codes with $R < \frac{2}{3}$. The measurement of improvement is defined as

$$IPR = \left(1 - \frac{\Delta_{NBP-BP}}{\Delta_{MS-BP}}\right) \times 100\%,$$

where Δ_{NBP-BP} is the difference of the minimum working SNR (SNR_{min}) between these normalized-BP algorithms and BP-FP, which results from the approximation inaccuracy and the quantization noise. Similarly, Δ_{MS-BP} is that between min-sum algorithm and BP-FP. For $R = \frac{1}{4}$ code that should work in low SNR condition, there is no suitable β in the set $\{\frac{1}{8}, \frac{2}{8}, \ldots, 1\}$ for the fixed-β approach, leading to $IPR = 0$. On the contrary, all the other dynamic normalizations in this case can still compensate about 40% SNR loss. The average

Table 4.2: Parameters of fixed-β and adaptive-β approaches

Rate	Fixed-β		Single-β			Double-β						
	β	$(a.b)$	β_{m_1}	β_{m_2}	$(a.b)$	β_{11}	β_{12}	T_1	β_{21}	β_{22}	T_2	$(a.b)$
$\frac{1}{4}$	0.875	(3.2)	0.75	1.0	(3.2)	0.5	0.75	0.5	1.0	1.0	—	(3.2)
$\frac{1}{3}$	0.875	(2.3)	0.75	1.0	(2.3)	0.625	0.75	0.625	0.875	1.0	2.0	(2.3)
$\frac{2}{5}$	0.875	(2.3)	0.625	1.0	(2.3)	0.5	0.75	1.25	0.75	1.0	1.25	(2.3)
$\frac{1}{2}$	0.75	(3.2)	0.625	0.875	(2.3)	0.625	0.875	1.5	0.75	0.875	1.625	(2.3)
$\frac{3}{5}$	0.75	(3.2)	0.625	0.875	(2.3)	0.5	0.75	2.0	0.75	1.0	2.0	(2.3)

Table 4.3: Parameters of the Annealing Adaptive-β approaches

Rate	Single-β				Double-β							
	β_{m_1}	β_{m_2}	L	$(a.b)$	β_{11}	β_{12}	T_1	β_{21}	β_{22}	T_2	L	$(a.b)$
$\frac{1}{4}$	0.75	1.0	3	(3.2)	0.375	0.5	0.5	0.75	0.75	—	3	(3.2)
$\frac{1}{3}$	0.625	0.875	2	(2.3)	0.625	0.75	2.0	0.75	0.875	1.5	3	(3.2)
$\frac{2}{5}$	0.625	0.75	3	(3.2)	0.5	0.625	1.5	0.625	0.875	1.125	3	(2.3)
$\frac{1}{2}$	0.5	0.75	2	(2.3)	0.5	0.625	1.75	0.625	0.75	2.0	2	(2.3)
$\frac{3}{5}$	0.5	0.75	3	(2.3)	0.375	0.625	2.0	0.625	0.75	1.0	3	(2.3)

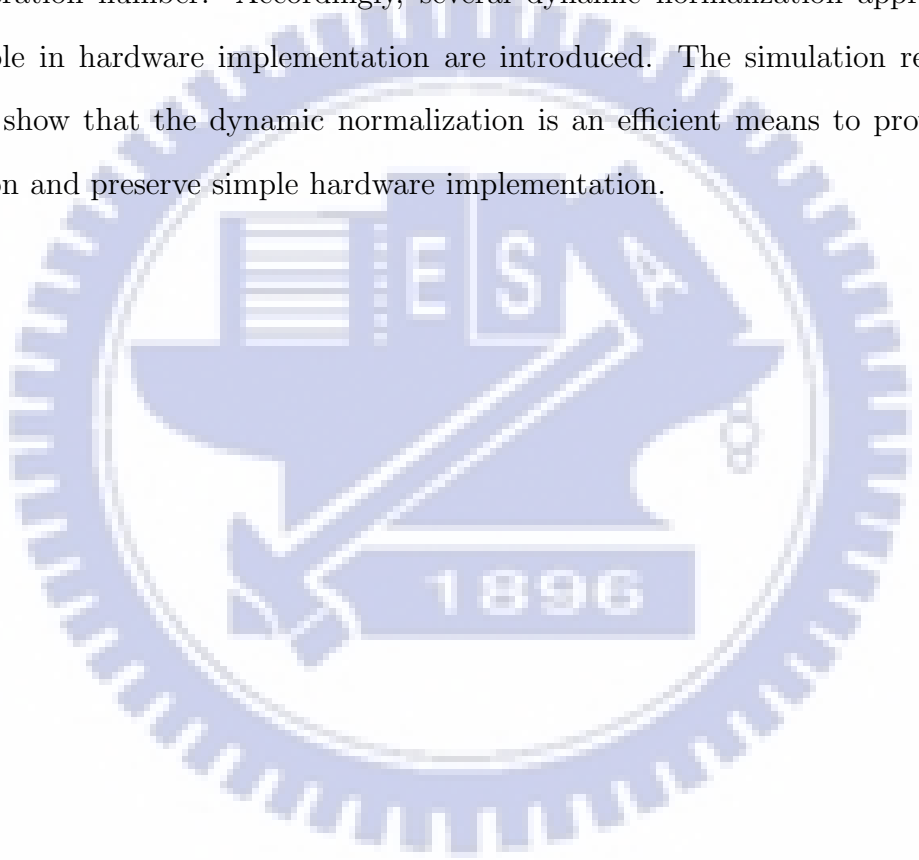
degradation $\bar{\Delta}_{NBP-BP}$ and the average improvement $\overline{IPR}$ are also given in Table 4.4. It shows that the double-β approach outperforms the others on average. The average SNR loss, $\bar{\Delta}_{NBP-BP}$, is reduced to 0.2dB while $\bar{\Delta}_{NBP-BP}$ of the fixed-β approach is 0.5dB. The average improvement of double-β approach is 72.9%, which is more than twice averaged IPR of the fixed-β approach.

Table 4.4: Comparisons of different normalization approaches

Rate	Measure	BP-FP	Min-Sum	Fixed-β	Annealing Single-β	Annealing Double-β	Single-β	Double-β
$\frac{1}{4}$	$\mathrm{SNR}_{\mathrm{min}}$(dB)	-2.55	-2.05	-2.05	-2.25	-2.25	-2.2	-2.25
	$IPR(\%)$	100	NA	0	44.4	44.4	33.3	44.4
$\frac{1}{3}$	$\mathrm{SNR}_{\mathrm{min}}$(dB)	-1.35	-0.6	-0.9	-1.0	-1.0	-1.05	-1.1
	$IPR(\%)$	100	NA	40.0	53.3	53.3	60.0	66.7
$\frac{2}{5}$	$\mathrm{SNR}_{\mathrm{min}}$(dB)	-0.45	0.55	0.25	0.0	-0.1	-0.2	-0.2
	$IPR(\%)$	100	NA	30.0	55.0	60.0	75.0	75.0
$\frac{1}{2}$	$\mathrm{SNR}_{\mathrm{min}}$(dB)	0.9	1.7	1.3	1.1	1.1	1.05	0.95
	$IPR(\%)$	100	NA	50.0	75.0	75.0	81.3	93.4
$\frac{3}{5}$	$\mathrm{SNR}_{\mathrm{min}}$(dB)	2.15	3.15	2.65	2.45	2.4	2.35	2.3
	$IPR(\%)$	100	NA	50.0	70.0	75.0	80.0	85.0
Average	$\bar{\Delta}_{NBP-BP}$	0	0.81	0.5	0.32	0.29	0.25	0.2
	$\overline{IPR}(\%)$	100	0	34	59.54	61.54	65.92	72.9

4.5 Summary

In this chapter, compensation schemes for approximation loss from Log-BP to min-sum algorithms are discussed. The correction factors of normalized min-sum algorithm are shown to be data-dependent. Based on order statistics and density evolution, the normalization factors can be described as a function of channel statistics (ex. SNR), decoder input, and decoding iteration number. Accordingly, several dynamic normalization approaches that are applicable in hardware implementation are introduced. The simulation results based on DVB-S2 show that the dynamic normalization is an efficient means to provide precise compensation and preserve simple hardware implementation.



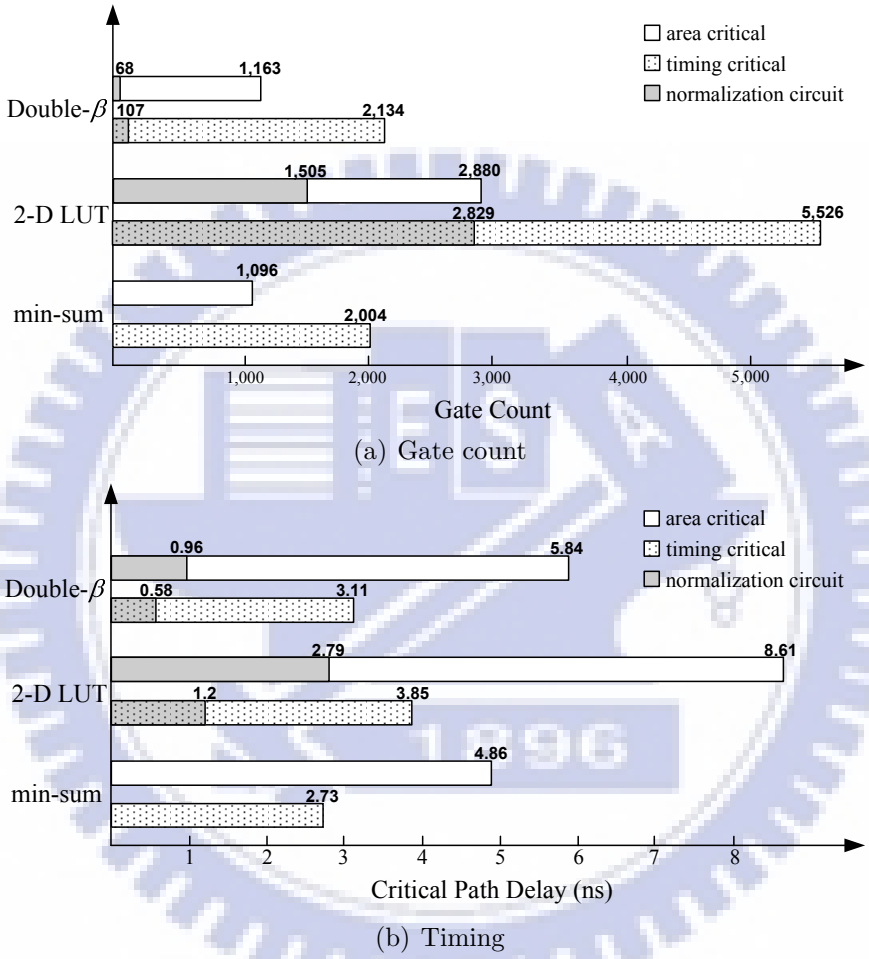


Figure 4.10: Implementation results (one check node unit) of the 2-D LUT, double-β approach, and min-sum algorithm for rate $\frac{3}{5}$, 64,800-bit LDPC code. The gray portion is the overhead introduced by the normalization circuit.

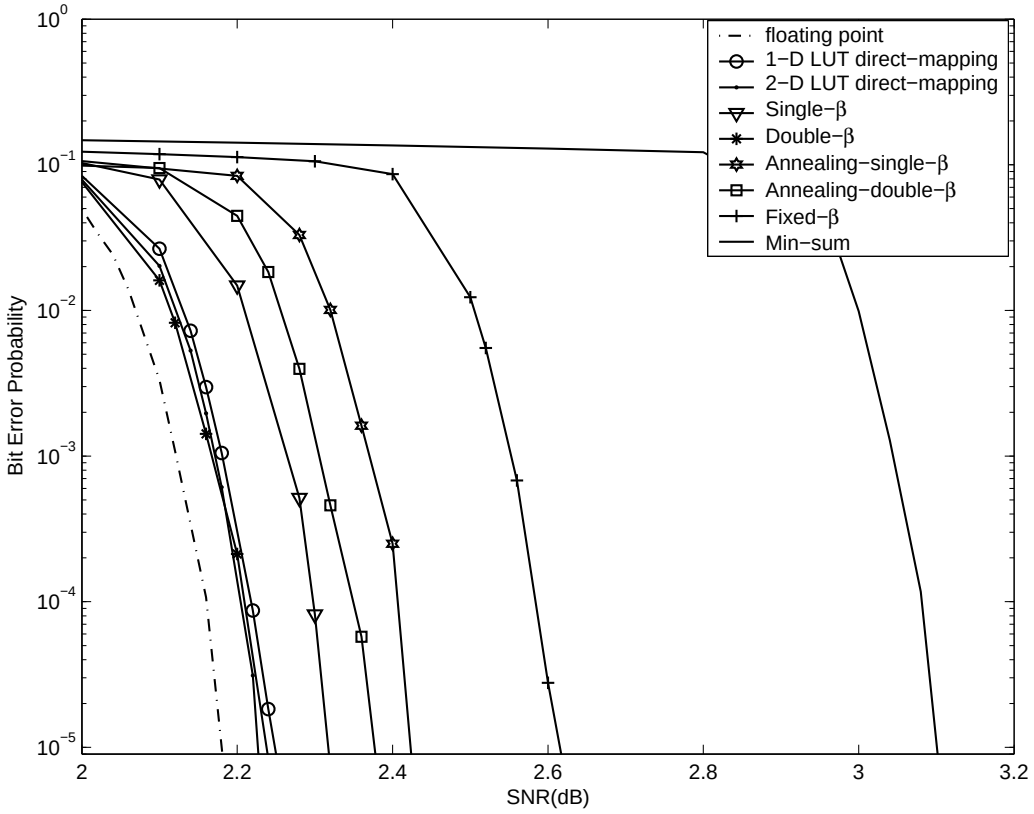


Figure 4.11: BER comparisons for rate $\frac{3}{5}$, 64800-bit LDPC with different normalizing techniques.

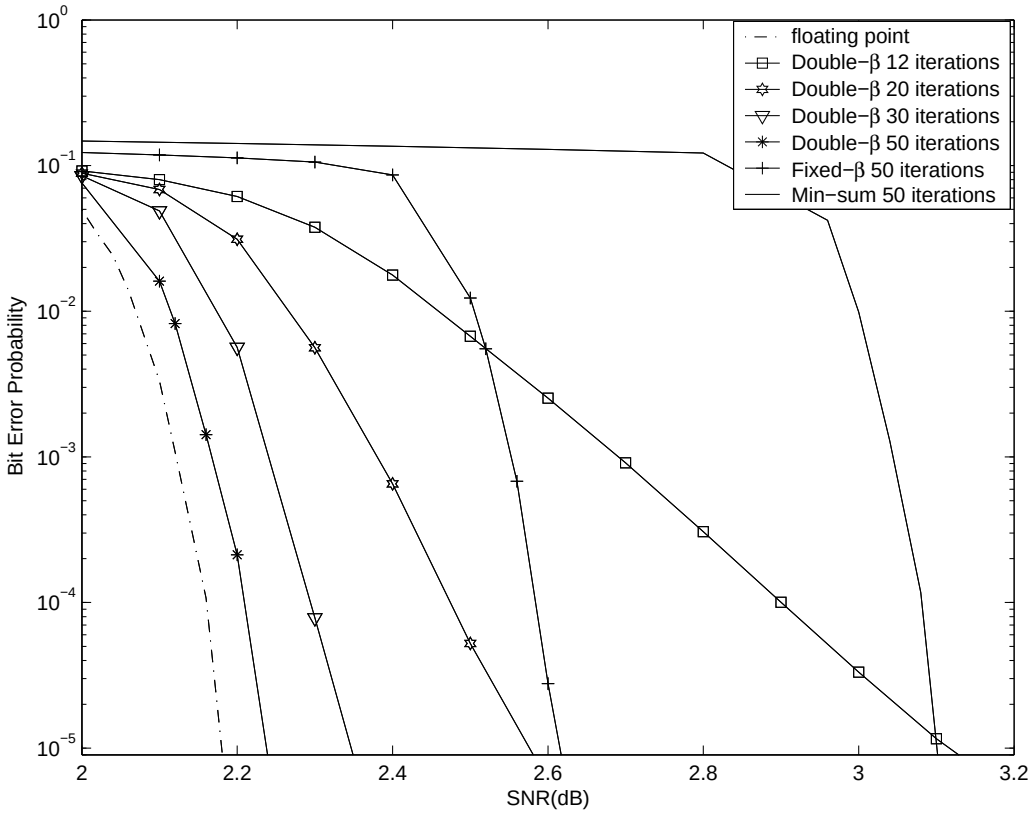


Figure 4.12: Comparisons of maximum decoding iterations for the rate $\frac{3}{5}$, 64800-bit LDPC code applied with different normalizing techniques. The simulation parameters and finite-precision message formats can be referred to Table 4.2.

Chapter 5

Channel-Coded MIMO Receiver

In Chapter 3, sphere decoding algorithm has been shown to be an efficient and applicable approach to realize ML detection for MIMO systems, and several techniques can be applied to further improve the computational efficiency. Combined with channel coding scheme, the additional coding gain allows the system work better in lower SNR environment. Instead of *hard-decision* inputs, many advanced channel coding schemes, turbo codes [6] or low-density parity check codes [7, 9] for instance, require the received data to have probabilistic information as soft value inputs. The sphere decoding algorithms introduced in Chapter 3 should be modified to generate the *soft values* (probabilistic information), and consequently *list sphere decoding algorithms* can be employed.

Modified from a sphere decoder, a list sphere decoder (LSD) performs almost the same operations but generates different output format. Not only the best guess of ML solution, a *candidate list* containing other symbols which have high probabilities of being ML solution is also delivered for computing the probabilistic information.

In the follow-up chapter, derivation of soft values from a list sphere decoder will be introduced first. Under message-passing decoding, the influences of soft value generation schemes are discussed, and low-complexity techniques for performance improvement will be proposed.

5.1 List Sphere Decoding Algorithm

Figure 5.1 illustrates soft-output MIMO detection realized by a list sphere decoder (LSD) where $\mathcal{L}$ is the candidate list of size $|\mathcal{L}|$. As the figure shows, an LSD primarily comprises two parts, candidate list generation and soft value generation. Generally, the candidate list generation can be realized by the sphere decoders introduced in Chapter 3, or by various sequential detection schemes [38,79]. The list size $|\mathcal{L}|$ dominates the computation complexity of the detector, thus it can be usually regarded as a parameter determined at design time.

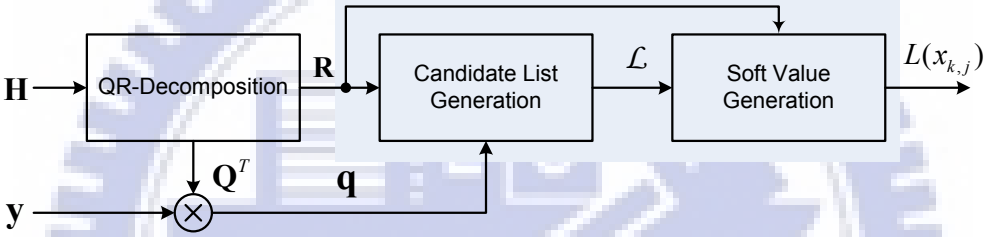


Figure 5.1: Soft-Output MIMO Detector.

5.1.1 Candidate List Generation and Soft Value Generation

An LSD differs from the conventional sphere decoder in the output format and the number of the outputs. A candidate list will be generated and the soft values are computed accordingly. If depth-first search is applied, the radius updating strategy needs to be modified as well. Since the sphere decoder is required to generate a candidates list with size $|\mathcal{L}|$, the radius will be fixed until the list is full. When the number of the retained paths exceeds $|\mathcal{L}|$, the path with the largest PED will be excluded from $\mathcal{L}$, and the radius will be updated to the currently largest PED in $\mathcal{L}$. Therefore, sorting for the maximum PED is performed whenever a new candidate is added into the list. As a result, sorting and radius-updating strategy

dominate the computation complexity. For breadth-first search with radius constraints, the same radius-updating philosophy should be employed. Considering constant decoding speed and computation, K-best algorithm is inherently suitable for candidate list generation. Moreover, the sorted K-best PEDs at the sphere decoder output can further reduce some computations in soft value generation. However, achieving low-error-rate and low-error-floor still requires a large K value.

For binary data, log likelihood ratio (LLR) is one of the most common description of the probabilistic information for the received data. The LLR of the bit $x_{j,k}$ is defined by its *a posteriori* probabilities, which is

$$\begin{aligned} L(x_{k,j}) &= \log \frac{Pr(x_{k,j}=0|\mathbf{y})}{Pr(x_{k,j}=1|\mathbf{y})} \\ &= \log \frac{Pr(x_{k,j}=0)}{Pr(x_{k,j}=1)} + \log \frac{Pr(\mathbf{y}|x_{k,j}=0)}{Pr(\mathbf{y}|x_{k,j}=1)}. \end{aligned} \tag{5.1}$$

The first term in (5.1) is the a priori information. This term is zero for ML detection or can be computed by the extrinsic information provided by the channel decoder in an iterative detection-decoding process [39]. Let $\mathcal{M}(\cdot)$ denote the M-PAM mapping function such that $s_k = \mathcal{M}(x_{k,1}, x_{k,2}, \ldots, x_{k,M_c})$. With Gaussian noise assumption, the second term in (5.1) can be computed by

$$\log \frac{Pr(\mathbf{y}|x_{k,j}=0)}{Pr(\mathbf{y}|x_{k,j}=1)}$$

$$= \log \frac{\sum_{\mathbf{s}' \in \mathbf{\Omega}_{j,0}} Pr(\mathbf{y}|\mathbf{s}')}{\sum_{\mathbf{s}' \in \mathbf{\Omega}_{j,1}} Pr(\mathbf{y}|\mathbf{s}')} \tag{5.2}$$

$$\approx \frac{1}{2\sigma^2}\left(\min_{\mathbf{s}' \in \mathbf{\Omega}_{j,1}} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 - \min_{\mathbf{s}' \in \mathbf{\Omega}_{j,0}} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 \right), \tag{5.3}$$

$$\approx \frac{1}{2\sigma^2}\left(\min_{\mathbf{s}' \in \mathbf{\Omega}_{j,1} \cap \mathcal{L}} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 - \min_{\mathbf{s}' \in \mathbf{\Omega}_{j,0} \cap \mathcal{L}} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 \right), \tag{5.4}$$

where σ^2 is the noise variance, and $\mathbf{\Omega}_{j,b}$ is the the set of all $\mathbf{s}'$ having $x_{k,j} = b$ for $b = 0, 1$. When preprocessing is performed, that is, $\mathbf{q} = \mathbf{Q}^T\mathbf{y}$ and $\mathbf{H} = \mathbf{QR}$, (5.4) will be

$$\frac{1}{2\sigma^2}\left(\min_{\mathbf{s}'\in\mathbf{\Omega}_{j,1}\cap\mathcal{L}} \|\mathbf{q}-\mathbf{R}\mathbf{s}'\|^2 - \min_{\mathbf{s}'\in\mathbf{\Omega}_{j,0}\cap\mathcal{L}} \|\mathbf{q}-\mathbf{R}\mathbf{s}'\|^2\right). \tag{5.5}$$

5.1.2 Dynamic Compensation

In order to further improve the approximation accuracy for the channel decoder soft inputs, an additive correction term that dynamically compensates the loss from (5.3) to (5.5) can be introduced.

Let n_0 and n_1 denote the sizes of $\mathbf{\Omega}_{j,0} \cap \mathcal{L}$ and $\mathbf{\Omega}_{j,1} \cap \mathcal{L}$ respectively, and $n_0 + n_1 = |\mathcal{L}|$. Moreover, let

$$m_0 = \min_{\mathbf{s}'\in\Omega_{j,0}} \|\mathbf{q}-\mathbf{R}\mathbf{s}'\|^2 \tag{5.6}$$

and

$$m_1 = \min_{\mathbf{s}'\in\Omega_{j,1}} \|\mathbf{q}-\mathbf{R}\mathbf{s}'\|^2. \tag{5.7}$$

Then we can express (5.3) as follows:

$$\begin{aligned}
\log\frac{\sum_{\mathbf{s}'\in\mathbf{\Omega}_{j,0}} Pr(\mathbf{y}|\mathbf{s}')}{\sum_{\mathbf{s}'\in\mathbf{\Omega}_{j,1}} Pr(\mathbf{y}|\mathbf{s}')} &= \log\frac{\sum_{\mathbf{s}'\in\mathbf{\Omega}_{j,0}} Pr(\mathbf{q}|\mathbf{s}')}{\sum_{\mathbf{s}'\in\mathbf{\Omega}_{j,1}} Pr(\mathbf{q}|\mathbf{s}')} \\
&= \frac{(m_1-m_0)}{2\sigma^2} + \log\frac{(1+\sum_{i=1}^{n_0-1} e^{\frac{-1}{2\sigma^2}(a_i-m_0)})}{(1+\sum_{i=1}^{n_1-1} e^{\frac{-1}{2\sigma^2}(b_i-m_1)})} \qquad (5.8) \\
&\le \frac{1}{2\sigma^2}\left(m_1 - m_0 + \log\frac{n_0}{n_1}\right), \qquad (5.9)
\end{aligned}$$

where $\{m_0, a_1, a_2, ..., a_{n_0-1}\} = \{T(\mathbf{s}')\}|\forall\mathbf{s}' \in \mathbf{\Omega}_{j,0} \cap \mathcal{L}\}$, $\{m_1, b_1, b_2, ..., b_{n_1-1}\} = \{T(\mathbf{s}')\}|\forall\mathbf{s}' \in$

$\mathbf{\Omega}_{j,1} \cap \mathcal{L}\}$, are the path metric of the paths in $\mathbf{\Omega}_{j,0} \cap \mathcal{L}$ and $\mathbf{\Omega}_{j,1} \cap \mathcal{L}$. Note that

$$\log \frac{(1 + \sum_{i=1}^{n_0-1} e^{\frac{-1}{2\sigma^2}(a_i - m_0)})}{(1 + \sum_{i=1}^{n_1-1} e^{\frac{-1}{2\sigma^2}(b_i - m_1)})} \leq \log \frac{n_0}{n_1} \tag{5.10}$$

can be regarded as a correction term. Moreover, for sufficiently large list size,

$$\log \frac{n_0}{n_1} \approx \log \frac{Pr(x_j = 0)}{Pr(x_j = 1)}, \tag{5.11}$$

which is the intrinsic information required by an maximum *a posteriori* (MAP) detector. As a result, the correction term and the intrinsic information can be combined as

$$\beta \log \frac{1 + n_0}{1 + n_1} \triangleq \log \frac{(1 + \sum_{i=1}^{n_0-1} e^{\frac{-1}{2\sigma^2}(a_i - m_0)})}{(1 + \sum_{i=1}^{n_1-1} e^{\frac{-1}{2\sigma^2}(b_i - m_1)})} + \log \frac{Pr(x_j = 0)}{Pr(x_j = 1)}. \tag{5.12}$$

Notice that $\frac{n_0}{n_1}$ is modified to $\frac{1+n_0}{1+n_1}$ to avoid logarithm of zero or infinity. Ultimately, the soft value will be

$$L(x_{k,j}) \approx \frac{1}{2\sigma^2}\left(m_1 - m_0 + \beta \log \frac{1 + n_0}{1 + n_1}\right), \tag{5.13}$$

where β is a normalization factor, and $n_1 = |L| - n_0$. From (5.13), the compensation overhead resulted from the dynamic compensation $\beta \log \frac{1+n_0}{1+n_1}$ are one multiplication, two logarithms, and at most $|\mathcal{L}| + 1$ additions for accumulating n_0.

5.2 Augmented-List Sphere Decoding Algorithm

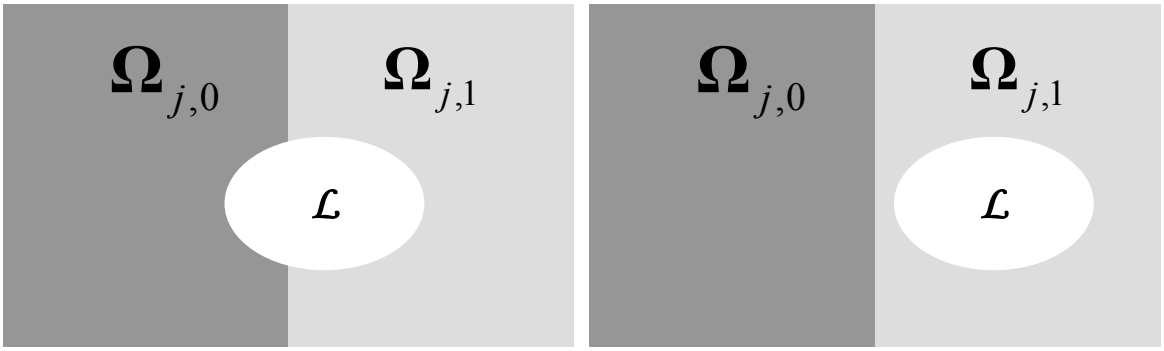
Let us take a closer look at (5.4) and (5.5). Chances are $\mathbf{\Omega}_{j,0} \cap \mathcal{L} = \emptyset$ or $\mathbf{\Omega}_{j,1} \cap \mathcal{L} = \emptyset$, it is impossible for us to find the minimizer in an empty set. In [39], it is suggested that the minima can be approximated by a predefined large constant in case of empty sets. Figure 5.2

is an illustrative examples for the empty set mentioned above. $\mathbf{\Omega}_{j,0}$ and $\mathbf{\Omega}_{j,1}$ equally partition the space of the valid constellation points. In Figure 5.2(a), we can always find minimizers in both $\mathbf{\Omega}_{j,0} \cap \mathcal{L}$ and $\mathbf{\Omega}_{j,1} \cap \mathcal{L}$ according to the given $\mathcal{L}$. An empty set is shown in Figure 5.2(b), and the list contains only the symbols in $\mathbf{\Omega}_{j,1}$. Then we can infer that $x_{k,j}$ has stronger confidence in 1, which corresponds to a smaller cost function, i.e. Euclidean norm. In other words, the weaker confidence in 0 should be represented by a large cost function. As a result, a large constant Euclidean norm is assigned.

5.2.1 Dealing with the Empty-Set Issue

Although we can assume $|\mathcal{L}|$ is large enough so the empty set rarely occurs, and [39] further suggested a list size larger than 512 is sufficient to maintain the desired error performance. However, $|\mathcal{L}| = 512$ is too large a list size for hardware implementation. Take 512-best algorithm for example, the average comparison operations per decoding layer will be $4608 \times M + \log_2 M$ (approximated by $512M \times \log_2(512M)$) for M-PAM mapping. Moreover, Figure 5.3 shows the rate of empty set versus the K value when K-best algorithm is employed for generating the candidate list. It is perceived that the empty set rate decreases much slower when $K \geq 64$ for 16-QAM and $K \geq 128$ for 64-QAM. In fact, this figure shows the improvement from enlarging the list size becomes limited eventually.

When approximated by a constant, the probabilistic information derived from (5.4) or (5.5) is equivalently added by an interference. Being the soft inputs to the subsequent channel decoder, the additional interference resulted from the approximation inaccuracy can hurt the error performance. Although the degradation can be mitigated by increasing the list size such that the probability of $\mathbf{\Omega}_{j,0} \cap \mathcal{L}$ (or $\mathbf{\Omega}_{j,1} \cap \mathcal{L}$) being an empty set reduces, the computation complexity in generating the candidate list also increases.



(a) $\mathbf{\Omega}_{j,1} \cap \mathcal{L} \neq \emptyset$ (b) $\mathbf{\Omega}_{j,1} \cap \mathcal{L} = \emptyset$

Figure 5.2: Illustration of the empty set issue

In [38,49], *path augmentation* techniques were proposed to expand $\mathcal{L}$ to a larger candidate list $\mathcal{L}'$ before soft value generation. Since $|\mathcal{L}'| > |\mathcal{L}|$, the probability of failing in finding the minimizers in $\mathcal{L}'$ is reduced. In general, the computation overhead resulted from list expansion is smaller as compared to directly generating a larger candidate list.

Although the path augmentation technique equivalently provides a larger list, we still have to estimate the minimas since $\mathbf{\Omega}_{j,0} \cap \mathcal{L}'$ or $\mathbf{\Omega}_{j,1} \cap \mathcal{L}'$ could still be an empty set. When this is the case, the simplest estimation of the minima is the the maximum path metric in $\mathcal{L}'$. This simple approach also applies to the conventional LSD where path-augmentation technique is not employed. That is, we can estimate the minima by the maximum path metric in $\mathcal{L}$.

5.2.2 Path Augmentation

Not only computation complexity, efficient path augmentation should also guarantee a low probability of failing in finding the minimizers. In the following, a path augmentation scheme is proposed. The candidate list $\mathcal{L}$ is expanded to distinct $\mathcal{L}_k$ for different $x_{k,j}$ such that we can always find the minimizers. Figure 5.4 shows the proposed augmented-list sphere

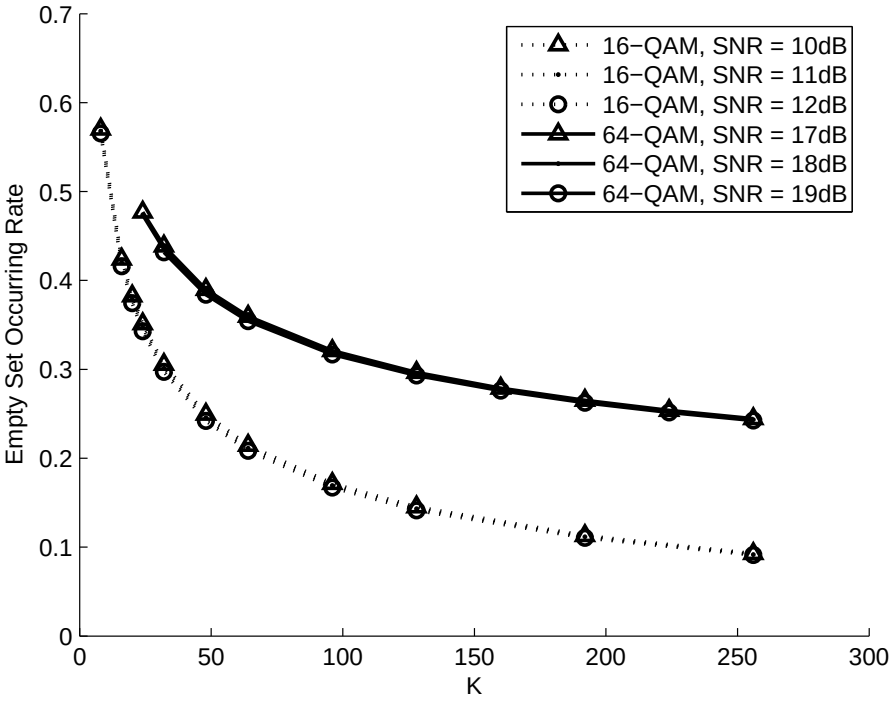


Figure 5.3: Empty set rates for 16-QAM and 64-QAM 4×4 system, the candidate list generation is realized by K-best algorithm.

decoder (A-LSD) in which the path augmentation can be treated as an enhancement; no modifications are required for the candidate list generation (sphere decoder) and the soft value generation.

When computing $L(x_{k,j})$, each path $\mathbf{s}'$ in $\mathcal{L}$ will be expanded to M paths by first duplicating $\mathbf{s}'$ $M-1$ times. Each the k-th element of the M identical paths is replaced by a distinct ω_j from $\mathbf{\Omega} = \{\omega_j | j = 0, 1, \ldots, M-1\}$, the M symbols of M-PAM constellation. This duplicating-and-replacing procedure continues until all the paths in $\mathcal{L}$ are examined. As a result, $\mathcal{L}$ is expended to $\mathcal{L}_k$ and $|\mathcal{L}_k| = M \times |\mathcal{L}|$. Although identical paths may be found in $\mathcal{L}_k$, $\mathbf{\Omega}_{j,0} \cap \mathcal{L}_k$ or $\mathbf{\Omega}_{j,1} \cap \mathcal{L}_k$ will never be empty sets since the augmented list contains all constellation points at the k-th layer. Figure 5.6 shows the augmented candidate list.

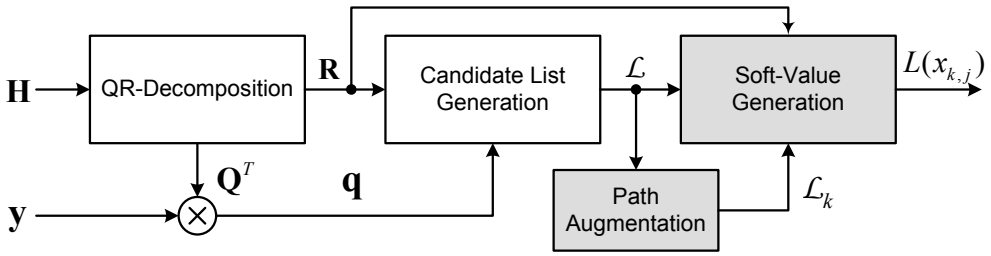


Figure 5.4: Augmented list sphere decoder.

Compared with Fig.5.2(b), the empty set in Figure 5.6(b) is now covered by the expanded $\mathcal{L}_k$. Besides, $\mathcal{L}$ is believed to be more reliable, and the augmented list is supposed to be reliable as well. It can be inferred that

$$\min_{\mathbf{s}' \in \mathbf{\Omega}_{j,0}} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 \approx \min_{\mathbf{s}' \in \mathbf{\Omega}_{j,0} \cap \mathcal{L}_k} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 \tag{5.14}$$

and

$$\min_{\mathbf{s}' \in \mathbf{\Omega}_{j,1}} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2 \approx \min_{\mathbf{s}' \in \mathbf{\Omega}_{j,1} \cap \mathcal{L}_k} \|\mathbf{y} - \mathbf{H}\mathbf{s}'\|^2. \tag{5.15}$$

Moreover, the path metric of the j-th expanded path from $\mathbf{s}'$ can be computed by

$$T(\mathbf{s}') + (\Delta_j \Sigma_k)^2 + 2\Delta_j \sqrt{\Sigma_k}, \tag{5.16}$$

where $\Delta_j = s_k - \omega_j$ for $j = 0, 1, \ldots, M-1$ and $\Sigma_k = \sum_{k'=k}^{2N_t} R_{k'k}$ is the k-th column summation of the channel matrix $\mathbf{R}$.

Figure 5.5 illustrates an example of the proposed path augmentation scheme for computing $L(x_{5,0})$ and $L(x_{5,1})$ in a 16-QAM 4×4 A-LSD. The equivalent 4-PAM 8-layered tree can be represented by an 8-stage trellis diagram. Each $\mathbf{s}'$ in $\mathcal{L}$ corresponds to a distinct path in the trellis. In this example, $\mathbf{s}' = \{+1, -1, -1, +1, +3, -1, -3, -1\}$, $M = 4$, and

$\mathbf{\Omega} = \{-3, -1, +1, +3\}$. $\mathbf{s}'$ is first expanded to the four distinct path that contains all constellation points of s_5 by duplicating-and-replacing procedure. Accordingly, $\mathbf{\Omega}_{0,0}$, $\mathbf{\Omega}_{0,1}$, and $\mathbf{\Omega}_{1,0}$, $\mathbf{\Omega}_{1,1}$ can be constructed.

As Figure 5.6 shows, the augmented list $\mathcal{L}_k$ equally partitions $\mathbf{\Omega}_{j,0}$ and $\mathbf{\Omega}_{j,1}$. Note that when the dynamic compensation $\beta \log \frac{1+n_0}{1+n_1}$ in (5.13) is applied, n_0 and n_1 will be computed from the original list $\mathcal{L}$.

Table 5.1: Average number of operations per bit for A-LSD

Operation	K-best Algorithm	Path Augmentation	Soft Value Generation	Dynamic Compensation
CMP	$\frac{MK}{M_c}\log_2(MK)$	0	γMK	0
MUL	$\frac{K}{2M_c}\sum_{i=1}^{2N_t}(2N_t - i + M)$	$\frac{2MK\gamma}{M_c}$	0	1
ADD	$\frac{K}{2M_c}\sum_{i=1}^{2N_t}(2N_t - i + M)$	$\frac{MK\gamma+N_t}{2N_tM_c}$	1	$\leq \gamma K + 1$
SQR	0	$\frac{MK\gamma}{M_c}$	0	0
SQRT	0	$\frac{K\gamma}{M_c}$	0	0
Log	0	0	0	2

Note: CMP, MUL, ADD, SQR, and SQRT stand for comparing, multiplication, addition, square, and square root operations, respectively.

5.2.3 Complexity Analysis

The aforementioned procedure needs to be performed $2N_t$ times for decoding $\mathbf{s}$, and (5.16) is the major computation overhead. Since Δ_j have limited values and ranges, they can be realized by a simple look up table or a decoder. Considering the overhead from path augmentation, $\mathcal{L}_k$ can be augmented partially; the soft values are generated by the $|\mathcal{L}| \times \gamma$, the most reliable paths for $0 < \gamma \leq 1$. The value γ provides a tradeoff between complexity and error performance.

TABLE 5.1 shows the average operation per bit for the proposed augmented-list sphere decoder where candidate list is generated by K-best algorithm. Note that the number of comparisons in the soft value generation increases since the $|\mathcal{L}|$ is now expanded to γMK.

5.3 Simulation Results

An LDPC-Coded 64-QAM 4×4 MIMO system was simulated. Randomly generated binary data are encoded by (1944, 972) LDPC code defined in IEEE 802.11n [17]. By direct spatial mapping the coded information is transmitted via an uncorrelated flat fading channel. The probabilistic information is generated by various list sphere decoders.

Subsequently, the LDPC codewords are decoded by Horizontal shuffled scheduling [76] combined with normalized min-sum algorithm. Constant normalization factor is 0.875. At most 10 iterations are performed to decode each LDPC codeword. For conventional list sphere decoders without path augmentation, the log belief-propagation (Log-BP) algorithm described in Chapter 4 is inapplicable due to the sensitivity to inaccurate input probabilistic information. Slight interference resulted from inaccurate soft value estimation can be amplified by the non-linear check node updating in the Log-BP decoding. As a result, the erroneous messages traverse and spread through the iterative process, leading to poor convergence and performance degradation. For fair comparisons among different list sphere decoders, only linear decoding, min-sum algorithm for example, is employed.

5.3.1 Error Performance

The bit error rate (BER) in Fig.5.7 shows the influences of the aforementioned path augmentation and dynamic compensation. Note that the minima in (5.5) will be set to D when $\mathbf{\Omega}_{j,0} \cap \mathcal{L} = \emptyset$ or $\mathbf{\Omega}_{j,1} \cap \mathcal{L} = \emptyset$. If dynamic compensation is applied, the normalization factor $\beta = 1$ are derived empirically. All the solid lines and dotted lines stand for the cases whether dynamic compensation (5.13) is applied.

First, let us compare the performance of the conventional list sphere decoders without compensation. As the figure shows, significant improvement is perceived when K, i.e. the

list size, increases. To achieve BER below 10^{-5}, K should be larger than 128, otherwise error floor arises. This coincides with Fig. 5.3 that the rate of the empty set decreases slowly after $k \geq 128$ for the 64-QAM curve.

Fig. 5.7 also shows that the dynamic compensation improves the error performance at low SNR region for all K values. However, the error flow presents still. Subsequently, comparing 64-best LSD with 64-best A-LSD, a significant improvement is perceived when path augmentation is applied. Not only the waterfall region, but the performance at the error-floor region improves. It shows that 64-best A-LSD even outperforms the conventional 128-LSD. In fact, it will be shown that the overhead resulted from path augmentation is far less than directly increasing K from 64 to 128. Furthermore, degradation of reducing γ from 1 to 0.25 is less than 0.1dB at the waterfall region. At the error-floor region, similar performance for $\gamma = 0.25$ and $\gamma = 1$ can be reached.

For other channel coding scheme, similar results can be obtained. Fig. 5.8 and Fig. 5.9 present the simulated bit error rates when the channel coding in the system is replaced by the (648, 324) LDPC code in in IEEE 802.11n [17] and the rate-$\frac{1}{3}$ 480-bit convolutional turbo code in IEEE 802.16e, which is also termed as WiMAX CTC. The LDPC code is decoded by the same algorithm as the (1944, 972) LDPC code; the turbo code is decoded by Max-log MAP algorithm [79]. Since the block length of the LDPC code and the turbo code are comparatively shorter, the waterfall region is less obvious. But the two figures both show that dynamic compensation and path augmentation provides significant improvements.

Although the results in Fig. 3.7 and Fig. 3.8 show imperceptible difference on the performance at SNR smaller than 20dB, which is the SNR region where the channel decoder works. However, Fig. 5.7 to Fig. 5.9 demonstrate the error performance is highly dependent on the list size K. The simulation results show that the K value of the conventional K-best

algorithm can be reduced to half, at least, when the proposed techniques are applied.

5.3.2 Influence of Candidate List Generation

Fig 5.10 illustrates the BER variation resulted from different candidate list generation schemes for an augmented list sphere decoder. Dynamic compensation is applied for all cases, and $\gamma = 1$ for all cases. When the path metric $\|\mathbf{q} - \mathbf{Rs}\|^2$ is replaced by

$$\sum_{i=1}^{2N_t} \left| q_i - \sum_{j=i}^{2N_t} R_{i,j} s_j \right|, \tag{5.17}$$

the PED computed at i-th layer is also modified to

$$T(\mathbf{s}^{(i)}) = \sum_{i'=i}^{2N_t} \left| q_i - \sum_{j=i}^{2N_t} R_{i,j} s_j \right|, \tag{5.18}$$

then (5.1) is further simplified. Compared with $\sum_{i'=i}^{2N_t} (q_i - \sum_{j=i}^{2N_t} R_{i,j} s_j)^2$, (5.18) has smaller dynamic data range. As a result, the retained K best paths are not necessarily the same as that derived from the conventional K-best algorithm, and a different candidate list can be deduced. Due to the smaller dynamic rage, the probability of eliminating ML path during breadth-first search increases. Compared with the 64-best A-LSD of Euclidean norm, 64-best A-LSD with (5.18) as the path metric has slight degradation at the waterfall region. However, thanks to the smaller data range, the simplification (5.17) and (5.18) lead to better performance at the error-floor region. The A-LSD output distribution in Fig. 5.11 illustrates the difference between the two path metric computation. The smaller data rage of the simplified path metric form means smaller variance, which will lead to faster convergence under message-passing algorithm. Besides, it is explained in Chapter 4 that the

normalization factor for min-sum decoding is a function of the LDPC decoder input. As a result, constant normalization, referred to fixed-β approach in Chapter 4, works better when input distribution has smaller data range.

Fig. 5.10 also shows the BER when K-best algorithm is replaced by the early-pruning K-best algorithm (EP-K-best) presented in Chapter 3, and the strict sorting is approximated by the coarse-granularity sorting for $L = 16$. Besides, the K values for the multi-K-best algorithm are further reduced to 16, 16, 16, 16, 32, 64, 64 (the first to the eighth layer) in order to reduce the sorting complexity. For BER above 10^{-4}, we can observe that absolute difference approximation results to about 0.6dB SNR degradation, which is almost the same SNR loss caused by the early pruning scheme. Then an extra 0.4dB loss is introduced by the multiple K reduction. However, the path metric definition impacts the BER at the error floor region, i.e. BER below 10^{-4}.

Although the pruning scheme guarantees a high probability of finding ML path, other potential candidates may be dropped by the radius constraints. Thus the deduced list size may be far less than K, leading to higher error floor as Euclidean norm is employed in computing the path metric. Fig. 5.12(a) shows the simulated probability of the list size resulted from EP-64-best algorithm, where 18.62% of the lists having size smaller than 63, and 11.76% are smaller than 48. Compared with Fig. 5.12(b), Only 4.39% of the deduced candidate list has size smaller than 48, providing a sufficiently large list for computing the soft values.

Fig. 5.10 to Fig. 5.12 briefly concludes the influence of candidate list generation. Simplified computations influence the waterfall region performance; the candidate list size impacts the error floor performance.

Table 5.2: Computation of LSD and A-LSD

Method	128-best LSD			64-best A-LSD ($\gamma = 1$)				64-best A-LSD ($\gamma = 0.25$)			
Oper.	SD	SVG+DC	Total	SD	PA	SVG+DC	Total	SD	PA	SVG+DC	Total
CMP	3414	128+0	3542 (100%)	1536	0	512+0	2048 (57.8%)	1536	0	128+0	1664 (46.9%)
MUL	1964	0+1	1965 (100%)	982	342	0+1	1325 (67.4%)	982	86	0+1	1069 (54.5%)
ADD	1964	1+129	2094 (100%)	982	171	1+65	1219 (58.2%)	982	43	1+17	1043 (49.8%)

Note: SD, PA, SVG are abreviated for sphere decoding, path augmentation, soft value generation.

Table 5.3: Average number of comparing (CMP) operations per bit for (1944, 972) LDPC coded 64-QAM 4×4 system

Candidate List Generation	Average CMP Operation @ SNR = 18dB	SNR (dB) @ BER = 10^{-4}	SNR (dB) @ BER = 10^{-6}
128-best LSD (Euclidean norm)	3542 (100%)	16.3	18.0
64-best A-LSD ($\gamma = 1$, Euclidean norm)	2048 (57.80%)	15.95	17.20
64-best A-LSD ($\gamma = 1$, absolute difference)	2048 (57.80%)	16.55	17.35
64-best A-LSD ($\gamma = 0.25$, Euclidean norm)	1664 (46.9%)	16.00	17.20
EP-64-best A-LSD (absolute difference, $L = 16$)	668 (18.9%)	16.55	17.45
EP-64-best A-LSD (Euclidean norm, $L = 16$)	636 (18.0%)	16.70	NA
EP-multi-K-best A-LSD (absolute difference, $L = 16$)	199 (5.6%)	16.95	17.7

5.3.3 Computation Complexity

So far, we have shown path augmentation scheme equivalently provides a larger candidate list. A 64-best A-LSD can even outperforms 128-best LSD. Not only the error performance, A-LSD also saves computation complexity. In Table 5.2, the computation complexity of various list sphere decoders are compared. Note that 128-best LSD is the reference. For 64-best A-LSD with $\gamma = 1$, at least 33% computations of the 128-best LSD are reduced. Furthermore, when γ is reduced to 0.25, about 50% of the computations can be saved. Since the comparing operation in sorting is most dominating, the sorting complexity is further compared and presented in Table 5.3. It is perceived that early-pruning scheme can further reduce the sorting complexity; 80% to 94% comparing operations can be reduced .

5.4 Summary

In this chapter, techniques to reduce computation complexity of list sphere decoders are presented. The path augmentation technique equivalently provides a larger and distinct list for each data bit, leading to reduced complexity and improved error floor performance. According to the simulation results, the K value of the conventional K-best algorithm can be reduced to half, at least. Besides, an additive correction term is introduced to dynamically compensate the simplification loss in computing the soft values. These scheme are applicable to many sphere decoding algorithms which are employed in generating the candidate list. Combined with the early-pruning K-best algorithm presented in Chapter 3, significant reduction in the computation complexity can be achieved.

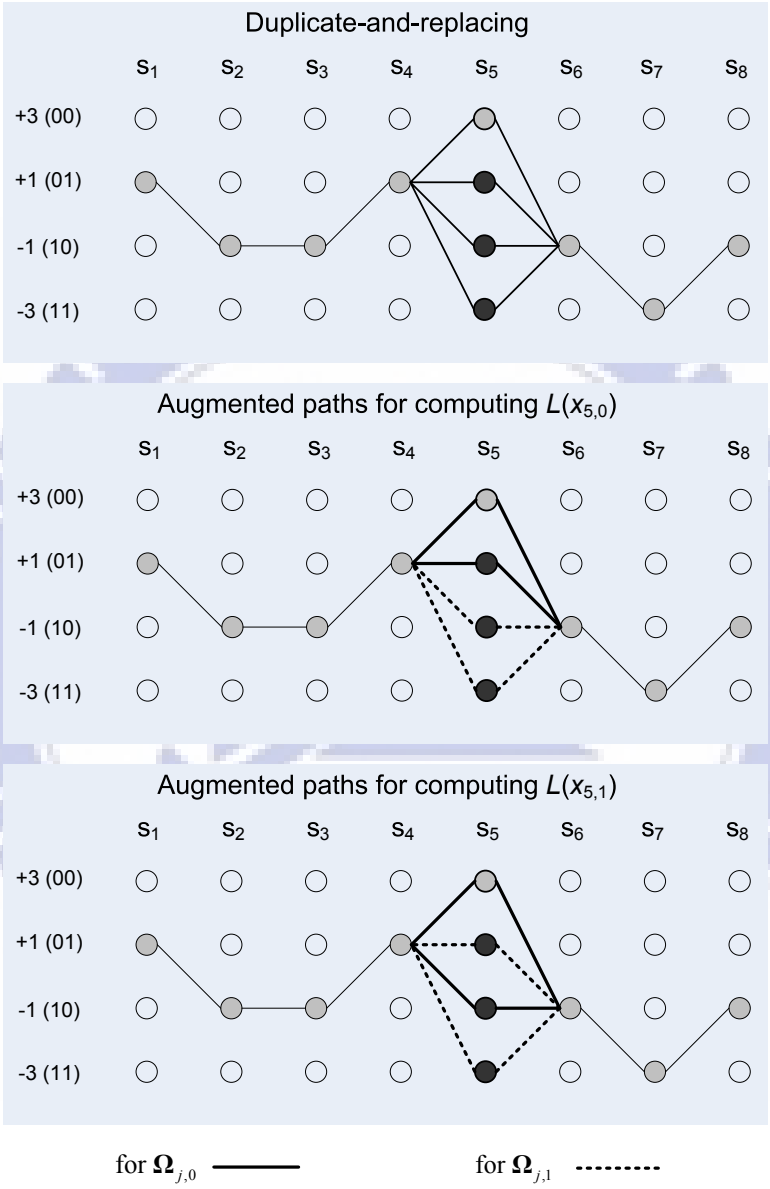
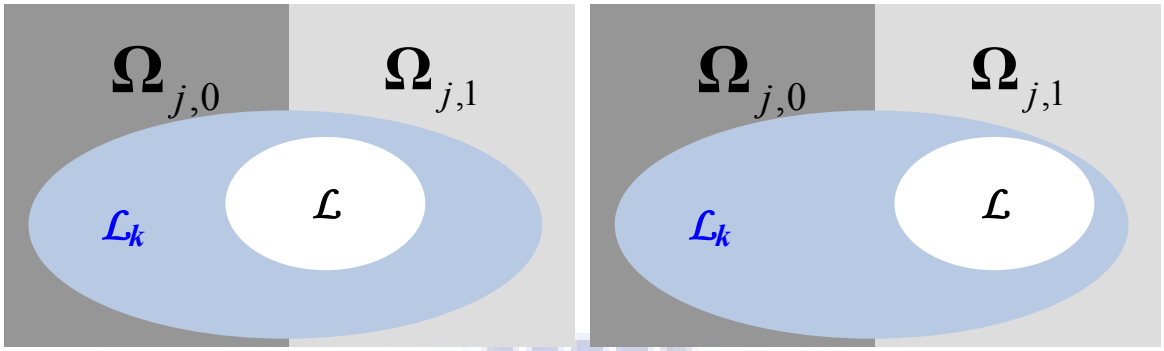


Figure 5.5: Path augmentation in a 16-QAM 4×4 A-LSD.



(a) $\mathbf{\Omega}_{j,1} \cap \mathcal{L} \neq \emptyset$ and $\mathbf{\Omega}_{j,1} \cap \mathcal{L}_k \neq \emptyset$ (b) $\mathbf{\Omega}_{j,1} \cap \mathcal{L} = \emptyset$ and $\mathbf{\Omega}_{j,1} \cap \mathcal{L}_k \neq \emptyset$

Figure 5.6: Path augmentation avoids finding minimas in an empty set.

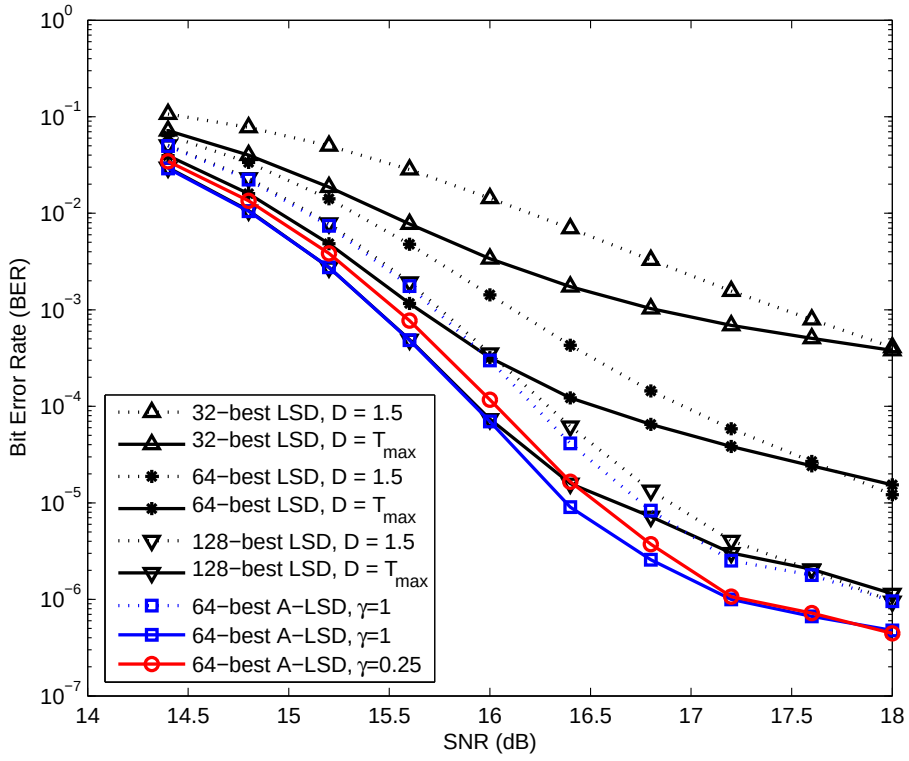


Figure 5.7: Simulated bit error rate of (1944,972) LDPC-coded 64-QAM 4×4 system.

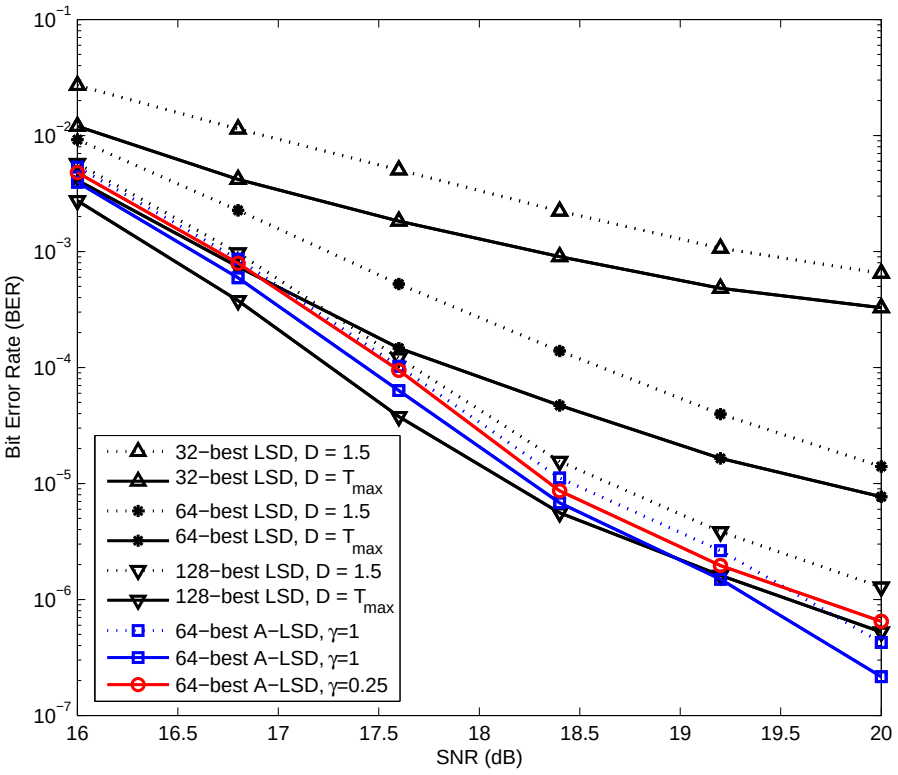


Figure 5.8: Simulated bit error rate of (648, 324) LDPC-coded 64-QAM 4×4 system.

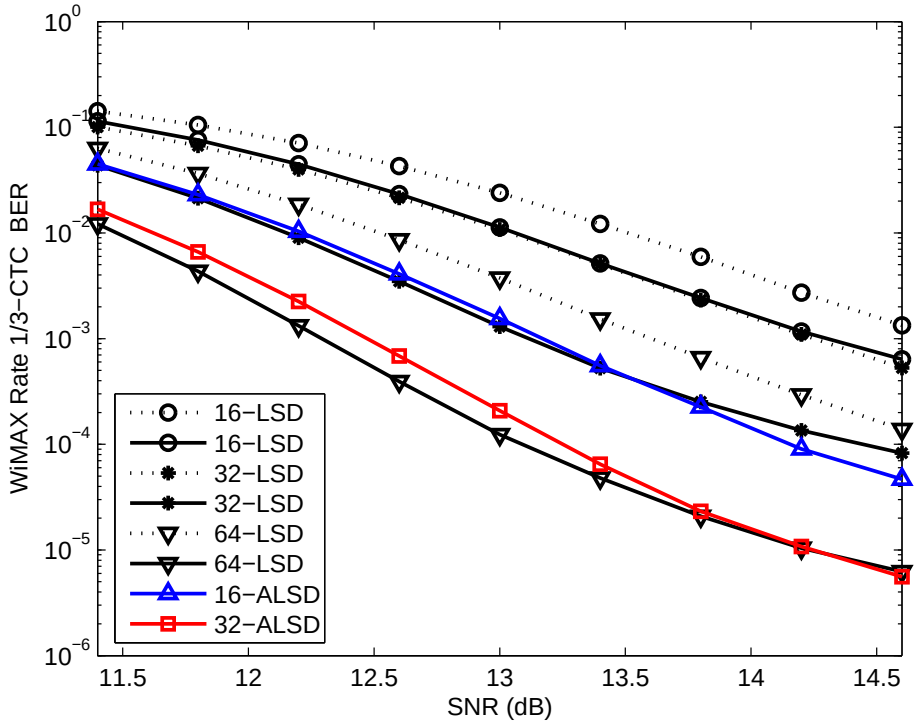


Figure 5.9: Simulated bit error rate of rate-$\frac{1}{3}$ convolutional-turbo-coded 64-QAM 4×4 system.

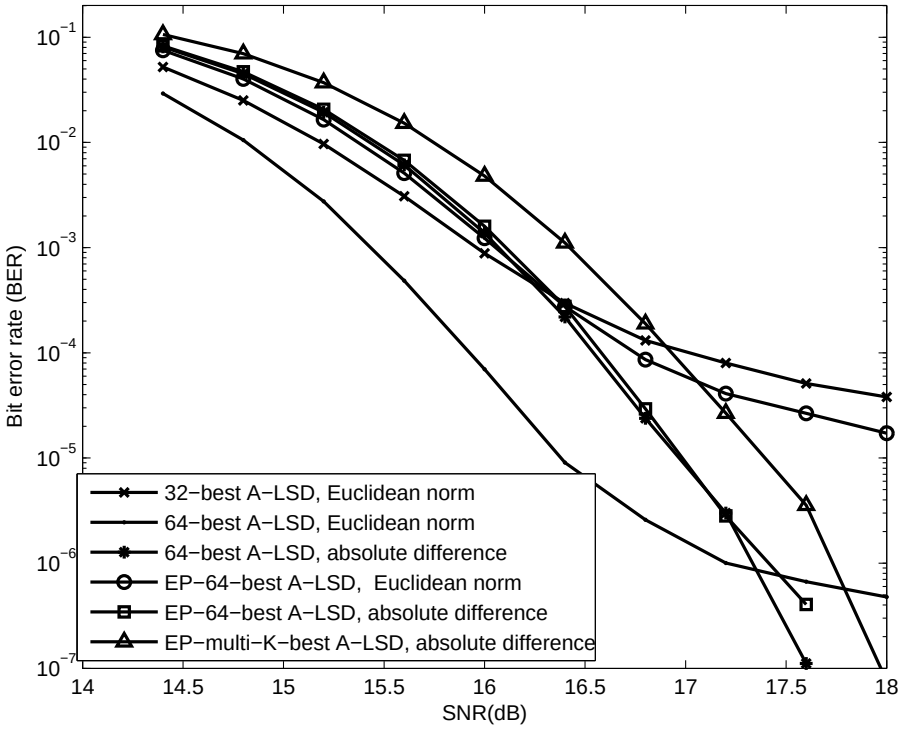


Figure 5.10: Simulated bit error rate of (1944,972) LDPC-coded 64-QAM 4×4 system. The candidate list of the A-LSD is realized by EP-K-best algorithm.

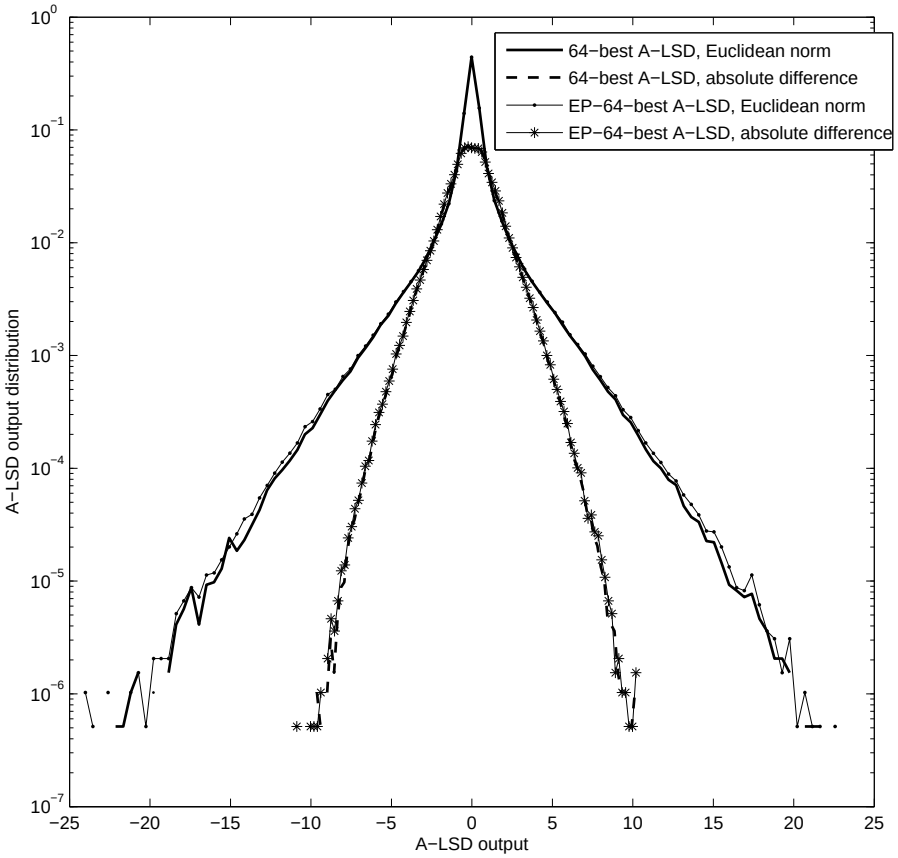
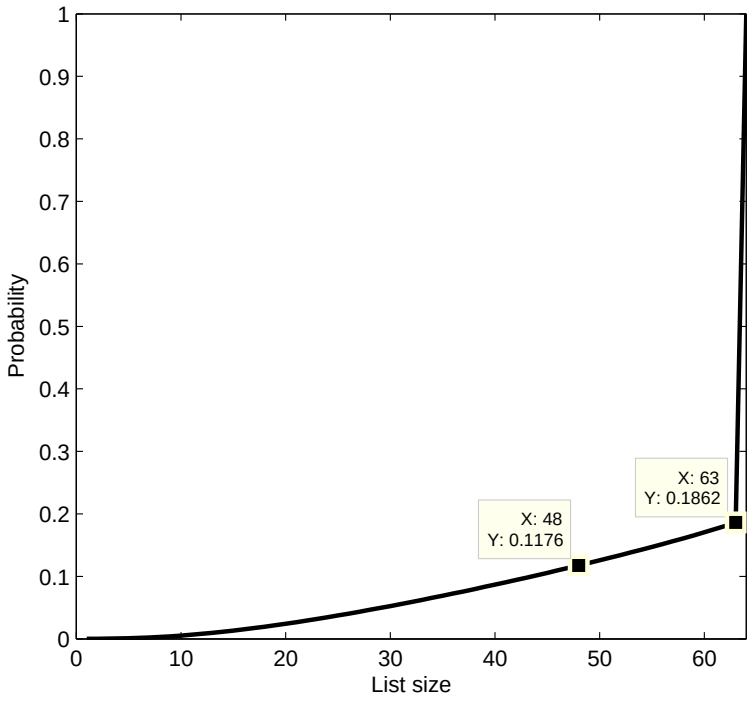
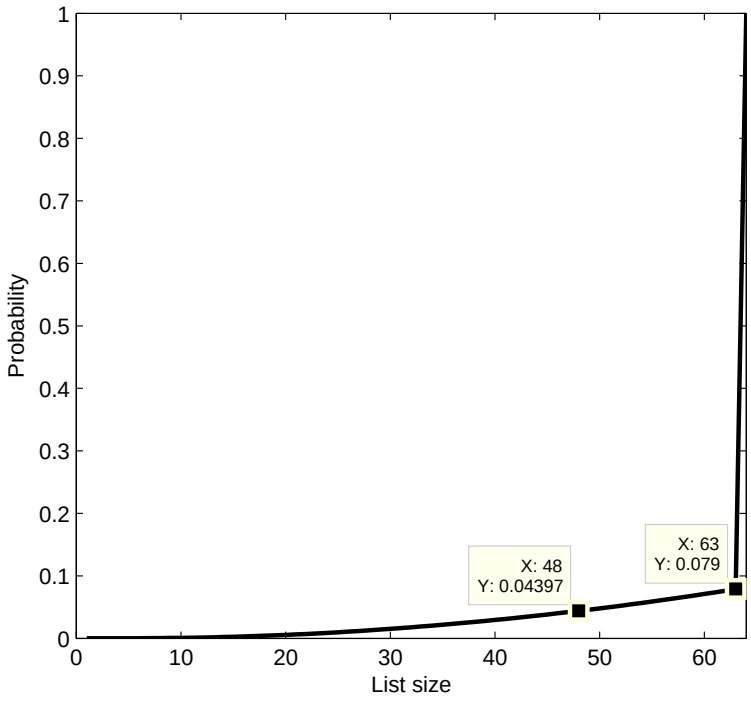


Figure 5.11: A-LSD output distribution.



(a) Euclidean norm



(b) Absolute difference

Figure 5.12: Simulated CDF of the list size by EP-64-best algorithm at SNR = 18dB.

Chapter 6

Conclusion

The thesis presents two essential parts in capacity-approaching MIMO receiver designs: sphere decoders and LDPC decoders that remarkably improve system performance but often demand costly hardware implementation. This work intends to reduce complexity in algorithm-leve and provide efficient solutions for the decoder.

6.1 Summary

Sphere decoding is one applicable realization for maximum likelihood (ML) signal detection in MIMO systems. Described as a closest-point-search problem, sphere decoding avoids exhaustive search in the entire signal space by confining a search range in a hypersphere. Sphere decoding can be further transformed to a tree-search problem and the search strategies can be categorized into depth-first and breadth-first. K-best algorithm is one popular realization of the latter by which constant computation and predictable complexity are guaranteed. Due to the fading phenomenon in propagation channels, a large K is required while considering the worst case scenario that the received signals are in deep fades. However, large K results in enormous computations, especially the sorting complexity. Consequently, we present an early pruning scheme that discard the less likely candidates during the search.

Applying to breadth-first sphere decoders, the proposed pruning scheme distinguishes the ML path from other paths by distinct radius constraints at each layer of the search tree.

Given the system model and the channel statistics, the pruning criterion, i.e. the radii, can be derived according to the desired error tolerance. Moreover, the expected computation complexity is analyzed. Although the computation is non-constant, the proposal ensures the manageable complexity by combining the pruning scheme with K-best algorithm. In fact, the algorithm will become conventional K-best algorithm when the number of the retained paths at each layer exceeds K, where the decoder reaches its lowest decoding speed. Besides, since the radii equivalently exhibits the dynamic ranges for the representation of ML path, the sorting complexity of the early-pruned K-best algorithm can be further reduced by coarse-granularity sorting approach. The presented analysis techniques also provide an approach to acquire parameters for multi-K-best algorithm by which each layer corresponds to a distinct K value. The lowest decoding speed can be improved by the multiple K's since some of the K values are smaller. Simulated in a 64-QAM 4×4 MIMO system, about twice improvement in the lowest decoding speed can be achieved when the early-pruned 64-best algorithm is modified to early-pruned multi-K-best algorithm. Moreover, both early-pruned 64-best and early-pruned multi-K-best algorithms can achieve similar error performance of the conventional 64-best algorithm. The degradation in SNR is almost imperceptible for BER above 10^{-5} while more than 90% computations are diminished.

After the MIMO detection process, the signals are passed to the channel decoder for error correction, and the LDPC code is one of the powerful and also popular coding techniques. Min-sum algorithm is often employed in implementing LDPC decoders for simplicity. The non-linear operations in the original decoding algorithm, Log-BP algorithm, is approximated by searching for the minima; however, significant performance loss may arise. Conventionally a constant offset or a normalization term can be applied to compensate the degradation, whereas in some cases constant factors can not accurately compensation the approximation

error. Thus, we investigate the parameters for normalized min-sum algorithms; the normalization factors can be represented as a function of decoder inputs and channel statistics. Consequently, dynamic normalization schemes are proposed. Based on order statistics and density evolution the data-dependent correction terms can be analyzed. The dynamic normalization preserves simple hardware implementation, and the resulted overheads in circuit complexity is less than 5% of the conventional min-sum algorithm. To reveal the effect of dynamic normalization, we apply the logn LDPC code defined in DVB-S2 system. Simulation results shown that the proposed techniques can provide as much as 1dB SNR improvement for min-sum algorithm.

Many of the research on LDPC decoding algorithms are based on AWGN channel model. In MIMO systems, however, the decoder convergence under iterative decoding is highly dependent on the input soft values. Sphere decoding algorithms are modified to list sphere decoding algorithm that generates a candidate list for computing the soft inputs to the LDPC decoder. We found that the list size impacts the error performance, and insufficient candidate can result in sever error floor. Since producing a large candidate list can be computation-demanding, a path augmentation technique is proposed to to enlarge the candidate list. As a result, computation complexity can be reduced while the error floor can be alleviated. The path augmentation technique can be regarded as performance enhancement and applied to many list sphere decoding algorithms. We simulated LDPC codes in IEEE802.11n system under 4×4 spatial multiplexing, the path augmentation scheme combined with the early-pruning multi-K-best algorithm can achieve the lower computation complexity as well as the lower error floor. About 94% complutation in the sorter can be saved as compared to the list sphere decoders based on 128-best algorithm.

6.2 Futurework

Complexity reduction techniques for sphere decoders are presented in this dissertation, however, the preprocessing part is not considered. Under finite precision data representation, various QR decomposition algorithms (Householder transformation, Givens Rotation, and Gram-Schmidt for example [54]), lead to different data stability. Moreover, similar complexity reduction techniques can apply to preprocessing and the path metric computation. The truncation error of the multiplications and additions can be analyzed and compensated accordingly, and some of the multiplications can be further replaced by the low-error reduced-width multipliers [80–86]. Furthermore, the K values of the early-pruning multi-K-best algorithm presented in Chapter 3 is determined empirically. Because they are determined according to the expected complexity, statistically derived K values should be feasible. On the other hand, computing the average complexity by (3.41) is very time-consuming when the sphere degree (n) is larger than 12. Besides, as described in Chapter 5, several factors impact the error floor. More complicated and realistic models should be considered, and approximations are required for more efficient and quick analysis.

For LDPC decoding, the analysis for the dynamic factors in Chapter 4 is based on standard Log-BP decoding algorithm, while the LDPC convergence behavior is different under shuffled decoding. Similar analyzing techniques still apply but require some modification.

References

[1] E. C. Shannon, "A mathematical theory of communication," *The Bell Lab Technical Journal*, vol. 27, pp. 379–423(Part–I), Jul. 1948.

[2] ——, "A mathematical theory of communication," *The Bell Lab Technical Journal*, vol. 27, pp. 623–656(Part–II), Jul. 1948.

[3] V. Poor, *An Introduction to Signal Detection and Estimation*. Springer, 1994.

[4] J. G. Proakis, *Digital Communications*. McGraw Hill Higher Education, 2000.

[5] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2001.

[6] C. Berrou and A. Glvieux, "Near optimum error correcting and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[7] R. G. Gallager, *Low-Density Parity-Check Codes*. MA: MIT Press, 1963.

[8] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[9] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

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

[11] S. Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 657–670, Feb. 2001.

[12] T. Richardson, M. A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 619–637, Feb. 2001.

[13] M. G. Luby, M. Mitzenmacher, M. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, pp. 585 – 598, Feb. 2001.

[14] J. L. Fan, *Constrained Coding and Soft Iterative Decoding.* Kluwer Academic Publishers, 2001.

[15] *Digital Video Bracasting (DVB) Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications*, ETSI Std. En 302 307, 2005.

[16] *Framing structure, Channel coding and modulation digital television terrestrial broadcasting system*, Academy of Broadcasting Planning Std. GB 20 600-2006, 2006.

[17] *Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput*, IEEE Std. P802.11n.D1.0, 2006.

[18] *Local and Metropolitan Area Network Part16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Draft*, IEEE Std. P802.16e.D9, 2005.

[19] *Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment to IEEE Std 802.3-2005*, IEEE Std. P802.3an, 2006.

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

[21] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, pp. 673–680, May 1999.

[22] X. Y. Hu, Eleftheriou, D. M. Arnold, and A. Dholakia, "Efficient implementation of the sum-product algorithm for decoding LDPC codes," in *IEEE GLOBECOM'01*, vol. 2, Nov. 2001, pp. 25–29.

[23] A. Anastasopoulos, "A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution," in *IEEE GLOBECOM'01*, vol. 2, Nov. 2001, pp. 1021 – 1025.

[24] H. S. Song and P. Zhang, "Very-low-complexity decoding algorithm for low-density parity-check codes," in *IEEE PIMRC'03*, vol. 1, Sep. 2003, pp. 161 – 165.

[25] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, pp. 406–414, Mar. 2002.

[26] N. Kim and H. Park, "Modified UMP-BP decoding algorithm based on mean square error," *IEE Electronics Letters*, vol. 40, pp. 816 – 817, Jun. 2004.

[27] H. Jun and K. M. Chugg, "Optimization of scaling soft information in iterative decoding via density evolution methods," *IEEE Trans. Commun.*, vol. 6, pp. 957 – 961, Jun. 2005.

[28] J. Chen and M. P. C. Fossorier, "Density evolution for two improved bp-based decoding algorithms of ldpc codes," *IEEE Communications Letters*, vol. 6, pp. 208 – 210, May 2002.

[29] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Y. Hu, "Reduced-complexity decoding of ldpc codes," *IEEE Trans. Commun.*, vol. 53, pp. 1288 – 1299, Aug. 2005.

[30] J. Zhang, M. Fossorier, D. Gu, and J. Zhang, "Improved min-sum decoding of ldpc codes using 2-dimensional normalization," in *IEEE GLOBECOM'05*, vol. 3, Nov. 2005, pp. 1187 – 1192.

[31] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics.* Prentice Hall, 1994.

[32] H. A. David and H. N. Nagaraja, *Order Statistics.* Wiley, 2003.

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

[34] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications.* New York: Cambridge University Press, 2005.

[35] B. Vucetic and J. Yuan, *Space-Time Coding.* West Sussex: Wiley, 2003.

[36] J. K. Winters, J. Salz, and R. D. Gitlin, "The impact of antenna diversity on the capacity of wireless communication systems," *IEEE Trans. Commun.*, vol. 42, pp. 1740 – 1751, Apr. 1994.

[37] H. Jafarkhani, *Space-Time Coding: Theory and Practice.* New York: Cambridge University Press, 2005.

[38] S. Baro, J. Baro, and M. Witzke, "Iterative detection of MIMO transmission using a list-sequential (LISS) detector," in *IEEE International Conference on Communications (ICC)*, vol. 4, May 2003, pp. 2653 – 2657.

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

[40] H. Vikalo, B. Hassibi, and T. Hassibi, "Iterative decoding for mimo channels via modified sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 2299 – 2311, Nov. 2004.

[41] A. Elkhazin, K. N. Plataniotis, and S. Pasupathy, "Reduced-dimension MAP turbo-BLAST detection," *IEEE Trans. Commun.*, vol. 54, pp. 108 – 118, Jan. 2006.

[42] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput*, vol. 44, pp. 463–471, Apr. 1985.

[43] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE. Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.

[44] M. Grotschel, L. Lovász, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization, 2nd ed.* New York: Springer-Verlag, 1993.

[45] M. Ajtai, "The shortest vector ptoblem in l_2 is NP-hard for randomized reductions," in *30th. Ann. ACM Symp. Theory Comput.*, 1998, pp. 10 – 19.

[46] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, no. 2, pp. 181–199, Sep. 1994.

[47] E. Agrell, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[48] K. W. Wong, C. Y. Tsui., R. S. K. Cheng, and W. H. Mow, "A vlsi architecture of a K-best decoding algorithm for mimo channels," in *Proc. IEEE Int Symp. Curcuits Stst.*, May 2002, pp. III–273–III–276.

[49] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for mimo detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[50] A. Burg, M. Borgmann, M. Wenk, and M. Zellweger, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1–12, Jul. 2005.

[51] S. Chen, T. Ahang, and Y. Xin, "Relax k-best mimo signal detector design and VLSI implementation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 3, pp. 328–337, Mar. 2007.

[52] C. Oestges and B. Clerckx, *MIMO Wireless Communications: From Real-World Propagation to Space-Time Code Design*. London: Academic Press, 2007.

[53] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*. New Jersey: Prentice-Hall, 2002.

[54] J. E. Gentle, *Matrix Algebra*. New York: Springer, 2007.

[55] S. W. K. K. P. Kim, "Log-likelihood-ratio-based detection ordering in V-BLAST," *IEEE Trans. Commun.*, vol. 54, pp. 302 – 307, Feb. 2006.

[56] S. Loyka and F. Loyka, "V-BLAST without optimal ordering: analytical performance evaluation for rayleigh fading channels," *IEEE Trans. Commun.*, vol. 54, pp. 1109 – 1120, June. 2006.

[57] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Golden, "V-blast: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *IEEE International Conference on Signals, Systems and Electronics (ISSSE)*, Sep. 1994, pp. 295–300.

[58] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 11 841 – 1852, Nov. 1999.

[59] G. J. Foschini, D. Foschini, M. J. Foschini, C. Foschini, and R. A. Foschini, "Analysis and performance of some basic space-time architectures," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 303 – 320, Apr. 2003.

[60] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. expected complexity," *IEEE Trans. Signal Process.*, vol. 53, pp. 2806 – 2818, Aug. 2005.

[61] ——, "On the sphere-decoding algorithm II. generalizations, second-order statistics, and applications to communications," *IEEE Trans. Signal Process.*, vol. 53, pp. 2819 – 2834, Aug. 2005.

[62] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley-Interscience, 1999.

[63] J. B. Anderson and S. Mohan, "Sequential coding algorithm: A survey and cost analysis," *IEEE Commun.*, vol. COM-32, no. 2, pp. 169–176, Feb. 1984.

[64] S. M. Razavizadeh, V. T. Vakili, and P. Azmi, "A new faster sphere decoder for MIMO systems," in *Proc. IEEE Int. Symp. Signal Processing and Information Technology (ISSPIT 2003)*, Dec. 2003, pp. 14–17.

[65] W. Zhao and G. B. Giannakis, "Sphere decoding algorithm with improved radius search," *IEEE Commun.*, vol. 53, no. 7, pp. 1104–1109, Jul. 2005.

[66] M. Bayat and V. T. Vakily, "Lattice decoding using accelerated sphere decoder," in *Proceedings of International Conference on Advanced communication Technology*, vol. 2, Feb. 2007, pp. 12–14.

[67] H. C. Chang, Y. C. Liao, and H. C. Chang, "Low-complexity prediction techniques of k-best sphere decoding for MIMO systems," in *IEEE Workshop on Signal Processing Systems (SiPS'07)*, Oct. 2007, pp. 45–49.

[68] J. Jie, C. Y. Tsui, and W. H. Mow, "A threshold-based algorithm and vlsi architecture of a k-best lattice decoder for mimo systems," in *IEEE International Symposium on Circuits and Systems (ISCAS 2005)*, May 2005, pp. 3359 – 3362.

[69] T. Cui, T. Ho, and C. Tellambura, "Statistical pruning for near maximum likelihood detection of MIMO systems," in *EEE International Conference on Communications(ICC'07)*, Jun. 2007, pp. 5462 – 5467.

[70] Q. L. amd Z. Wang, "Early-pruning k-best sphere decoder for mimo systems," in *IEEE Workshop on Sinal Processing Systems (SiPS'07)*, Oct. 2007, pp. 40 – 44.

[71] R. Gowaikar and B. Hassibi, "Statistical pruning for near-maximum likelihood decoding," *IEEE Trans. Signal Process.*, vol. 55, pp. 2661–2675, Jun. 2007.

[72] Y. S. Wu, Y. T. Liu, H. C. Chang, Y. C. Liao, and H. C. Chang, "Early-pruned k-best sphere decoding algorithm based on radius constraints," *to be presented in 2008 IEEE International Conference on Communications (ICC'08).*

[73] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 5, pp. 399–431, Sep. 1981.

[74] R. Diestel, *Graph Theory.* Springer, 2006.

[75] Y. C. Liao, C. C. Lin, H. C. Chang, and C. W. Liu, "Self-compensation technique for simplified belief-propagation algorithm," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 3061–3072, Jun. 2007.

[76] J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.*, vol. 53, p. 209V213, Feb. 2005.

[77] J. Zhang, , Y. Wang, M. P. C. Fossorier, and J. S. Yedidia, "Iterative decoding with replicas," *IEEE Trans. Inform. Theory*, vol. 53, pp. 1644–1663, May 2007.

[78] X. Wei and A. N. Akansu, "Density evolution for low-density parity-check codes under max-log-map decoding," *IEE Electronics Letters*, vol. 37, pp. 1125 – 1126, Aug. 2001.

[79] S. Lin and D. J. Costello, *Error Control Coding, Second Edition.* New Jersey: Prentice Hall, 2004.

[80] J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of lower-error fixed-width multipliers for dsp applications," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 6, pp. 836 – 842, Jun. 1999.

[81] L. D. Van, S. S. Wang, and W. S. Feng, "Design of the lower error fixed-width multiplier and its application," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 10, pp. 1112 – 1118, Oct. 2000.

[82] S. J. Jou, M. H. Tsai, and Y. L. Tsao, "Low-error reduced-width booth multipliers for dsp applications," *IEEE Trans. Circuits Syst. I*, vol. 50, no. 11, pp. 1470–1474, Nov. 2003.

[83] K. J. Cho, K. Lee, J. G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified booth multiplier," *IEEE Trans. VLSI Syst.*, vol. 12, no. 5, pp. 522–531, May 2004.

[84] L. D. Van and C. C. Yang, "Generalized low-error area-efficient fixed-width multipliers," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.

[85] T. B. Juang and S. F. Hsiao, "Low-error carry-free fixed-width multipliers with low-cost compensation circuits," *IEEE Trans. Circuits Syst. II*, vol. 52, no. 6, pp. 299–303, Jun. 2005.

[86] Y. C. Liao, H. C. Chang, and C. W. Liu, "Carry estimation for two's complement fixed-width multipliers," in *IEEE Workshop on Signal Processing Systems (SiPS'06)*, Oct. 2006, pp. 345 – 350.