



uOttawa

L'Université canadienne
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES**



**FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES**

Wei Yu

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE / DEGREE

Department of Electrical Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Optimum Power Allocation in V-Blast with High-Level Modulation

TITRE DE LA THÈSE / TITLE OF THESIS

S. Loyka

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

C. D'Amours

T. Kwasniewski

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

OPTIMUM POWER ALLOCATION IN V-BLAST WITH HIGH-LEVEL MODULATION

by

Wei Yu

The thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of
MASTER OF APPLIED SCIENCE
in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering,
University of Ottawa, Ottawa, Ontario, Canada

Copyright © Wei Yu, Ottawa, Canada, 2009



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-58190-2
Our file *Notre référence*
ISBN: 978-0-494-58190-2

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■+■
Canada

Abstract

Optimal ordering procedure, which is computationally complex in the conventional vertical Bell-labs layered space-time (V-BLAST) algorithm, can be simplified. Among those low-complexity alternatives to the optimal ordering, non-uniform power allocation is intuitively attractive. The unified analysis of power optimized zero forcing (ZF) VBLAST proposed by prior work ([32]) is extended from BPSK to high-level modulations (also known as high-order modulations) by introducing a generic SER approximation—average union bound (AUB)—for multi-path fading channels. This AUB model represents SER by using two parameters related to the constellation— D_s , the normalized minimum Euclidean distance squared and N_c , the average number of the nearest constellation neighbors. First of all, the validity of AUB is demonstrated. Consequently, general average block error rate (BLER) of ZF-VBLAST can be approximated by a closed-form expression of D_s and N_c for high-level modulation, the optimum power allocation solution are obtained at high and low SNR. The SNR gain and the robustness of this optimization are also rigorously defined and investigated in a generic manner. Additionally, the above performance variations related to the constellation size are discussed. At last, an efficient Monte Carlo simulation methodology, parallel Matlab®, is adopted for the simulations in this thesis. The observations of parallelism efficiency are summarized.

Acknowledgement

I wish to express my gratitude to my academic supervisor Dr. Sergey Loyka for his insightful guidance and thoughtful help. I would also like to thank my fellows in the group, Georgy and Yaobin for many constructive suggestions.

I thank my parents, my daughter Peiqi and my dear wife Wenyan here for their support and encouragement during my study course of this thesis.

Table of Contents

1. INTRODUCTION.....	1
1.1. MOTIVATION OF THE RESEARCH.....	1
1.2. CONTRIBUTION OF THIS THESIS.....	3
1.3. ORGNIZATION OF THE THESIS.....	5
2. LITERATURE REVIEW.....	7
2.1. SPATIAL MULTIPLEXING.....	9
2.2. SPATIAL DIVERSITY.....	12
2.3. SUMMARY.....	21
3. VBLAST ALGORITHM AND ITS PERFORMANCE.....	23
3.1. VBLAST ALGORITHM.....	23
3.2. OUTPUT SNR IN ZF-VBLAST STEPS.....	27
3.3. PERFORMANCE IN ZF-VBLAST STEPS.....	31
3.4. SUMMARY.....	35
4. AVERAGE UNION BOUND (AUB) OF SNR.....	36
4.1. BOUND OF INSTANTANEOUS SER.....	36
4.2. BOUND OF AVERAGE SER OVER RAYLEIGH FADING.....	39
4.3. BOUND OF SER OVER DIVERSITY RAYLEIGH CHANNELS.....	42
4.4. DISCUSSION.....	47
4.5. SUMMARY.....	48

5. OPTIMUM POWER ALLOCATION.....	50
5.1. BLOCK ERROR RATE OF VBLAST.....	50
5.2. OPTIMUM POWER ALLOCATION FOR VBLAST.....	52
5.3. SUMMARY.....	60
6. ROBUSTNESS OF THE OPTIMUM ALLOCATION.....	65
6.1. LOCAL ROBUSTNESS.....	66
6.2. GLOBAL ROBUSTNESS.....	68
6.3. SUMMARY.....	70
7. SNR GAIN.....	71
7.1. GENERAL BOUND OF SNR GAIN.....	71
7.2. SNR GAIN ASYMPTOTIC BEHAVIOR.....	72
7.3. SNR GAIN AT SMALL SNR REGION.....	73
7.4. SUMMARY.....	74
8. PARALLEL MC SIMULATION.....	76
8.1. PARALLELISM IN MATLAB® APPLICATION.....	76
8.2. EVALUATION OF PARALLELISM EFFICIENCY.....	80
8.3. SUMMARY.....	83
9. CONCLUSION.....	85
9.1. SMMARY OF THE THESIS.....	85
9.2. FUTURE RESEARCH.....	87

REFERENCE.....	89
APPENDIX A. MATHEMATICAL DERIVATION.....	98
A1. WEIGHTING FACTOR IN MMSE-VBLAST.....	98
A2. SIMPLIFIED UNION BOUND OF INSTANTANEOUS SER.....	100
A3. ASYMPTOTIC APPROXIMATION OF SER OVER DIVERSITY.....	102
A4. ASYMPTOTIC APPROXIMATION OF VBLAST BLER.....	104
A5. OPTIMUM POWER ALLOCATION COEFFICIENTS.....	105
A6. DERIVATION OF $\bar{P}_{ei}$ AROUND $\gamma_0 = 0$	108
A8. LOCAL ROBUSTNESS δ_i'	110
A9. TRUNCATED ASYMPTOTIC APPROXIMATION OF $\bar{P}_B(\alpha)$	111
APPENDIX B. PARALLEL MATLAB® TOOLKITS.....	113
B1. EVALUATION OF EXISTENT PARALLEL MATLAB® TOOLKITS.....	113
B2. GUIDE OF USING PMATLAB.....	117
B3. USAGE OF THE PARALLEL COMPUTING TOOLS FOR MATLAB 2007B.....	120
APPENDIX C. MATLAB® CODE.....	126
C1. FUNCTIONS.....	126
C2. MONTE CARLO (MC) SIMULATION.....	139
C3. PARALLEL MC SIMULATION.....	142

List of Acronyms

Acronym	Meaning
AOA	angle of arrival
AUB	average union bound
AWGN	additive white Gaussian noise
BER	bit error rate
BLER	block error rate
BPSK	binary phase shift keying
CDMA	code division multiple access
dB	decibel
D-BLAST	Diagonal Bell Laboratories Layered Space-Time
DF	decision feedback
EDGE	enhanced data rates for Global System for Mobile (GSM) evolution
FFT	fast Fourier transforms
GSM	Global System for Mobile
IEEE	Institute of Electrical and Electronics Engineers, Inc.
ISI	inter symbol interference
MC	Monte Carlo
ML	maximum likelihood
MIMO	multiple-input multiple-output
MISO	multiple-input single-output
MMSE	minimum mean-square error
MPSK	M-level phase shift keying
MQAM	M-level quadrature amplitude modulation
OFDM	orthogonal frequency-division multiplexing
PDF	probability density function
QPSK	quadrature phase shift keying
Rx	receiver
SER	symbol error rate

SQP	sequential quadratic programming
SIC	successive interference cancelling
SIMO	single-input multiple-output
SISO	single-input single-output
SNR	signal to noise ratio
STBC	space-time block codes
STTC	space-time trellis code
STC	space-time code
Tx	transmitter
TBER	total bit error rate
UMTS	Universal Mobile Telecommunications System
V-BLAST	Vertical Bell Laboratories Layered Space-Time
WLAN	wireless local area network
ZF VBLAST	zero-forcing VBLAST

List of Symbols and Notations

Notation	Meaning
T	transposition
$+$	conjugate transposition
$ \mathbf{a} $	Euclidean norm of vector $\mathbf{a}$
a_i	i -th component of vector $\mathbf{a} = [a_1, a_2, \dots, a_m]^T$
$\mathbf{a}_i$ or $(\mathbf{A})_i$	i -th column of matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m]$
$\mathbf{A}^{(i)}$	matrix $\mathbf{A}$ without i -th column $\mathbf{A}^{(i)} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{i-1}, \mathbf{a}_{i+1}, \dots, \mathbf{a}_m]$
$\mathbf{A}_{(i)}$	$\mathbf{A}_{(i)} = [\mathbf{a}_i, \mathbf{a}_{i+1}, \dots, \mathbf{a}_m]$

Symbol	Meaning	First appearance
m	number of transmit antennas	Section 2.1
n	number of receive antennas	Section 2.1
$\mathbf{s} = [s_1, s_2, \dots, s_m]^T$	Tx vector	(3.1)
$\mathbf{r} = [r_1, r_2, \dots, r_m]^T$	Rx vector	(3.1)
$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m]$	MIMO channel matrix, $\mathbf{h}_i$ is i -th column of $\mathbf{H}$	(3.1)
$\xi \in \mathcal{CN}(\mathbf{0}, \sigma_0^2 \mathbf{I})$	noise vector	(3.1)
σ_0^2	noise power	(3.1)
$\mathbf{h}_i'$	channel vectors after the optimal orderings	(3.5)
$\hat{s}_i'$	detected symbol at i -th step	(3.5)
$\mathbf{w}_i$	nulling vector	(3.10)
$P_e(\gamma_i)$	conditional Error Probability	Section 3.3
P_B	BLER (probability of having at least one error at the detected vector)	(3.22)
d_{min}	minimum Euclidean distance	Section 4.1
$D_s(d_{min})$	$D_s(d_{min}) = (d_{min}^2 / 4N_0\gamma_s)$, normalized minimum Euclidean distance squared	Section 4.1
$N_c(d_{min})$	average numbers of the nearest neighbors each constellation point has	Section 4.1
$\gamma_0 = 1/\sigma_0^2$	average per-Tx branch SNR for unoptimized V-BLAST	(4.1)

α_i	power allocation coefficient at i -th branch	(5.4)
λ	Lagrange multiplier for optimum power allocation derivation	(5.5)
$\delta = \left \frac{\Delta \bar{P}_B / \bar{P}_B}{\Delta \alpha_i / \alpha_i} \right $	robustness measure of optimum power allocation algorithm	(6.1)
G_{SNR}	SNR gain found from $\bar{P}_B(\alpha_{opt}) = \bar{P}_B(\alpha, \alpha, \dots, \alpha)$, $G_{\text{SNR}} = \alpha$	(7.1)

Chapter 1. Introduction

1.1. Motivation for the Research

Since the concept of multiple-input-multiple-output (MIMO) wireless communication was proposed in the mid- 1990s, it has been regarded as one of the most significant breakthroughs in achieving the Shannon channel capacity, which was established in 1948. In recent decades, numerous research efforts have focused on developing practical coding/decoding techniques to achieve the full MIMO capacities predicted by [20], [64] and [71]. Among these efforts, the Vertical Bell Laboratories Layered Space-Time (V-BLAST) ([19], [72]) algorithm is comparatively effective and has inspired many related research activities. The major obstacle in implementing V-BLAST is its computationally demanding optimum ordering procedure, which is crucial for diminishing error propagation introduced by the successive interference cancelling (SIC) technology. Besides those works that address the simplification of the pseudo-inverse operations on the channel matrix (e.g., [6] and [33]), alternatives to the optimum ordering have also been thoroughly studied recently ([11], [12], [28], [32], [46], [53], [69] and [70]). These alternatives also include a variant of maximum likelihood (ML) decoding and optimum transmission power allocations. Among them, the power-optimized V-BLAST without ordering has been shown to be an alternative to the optimum ordering with roughly the same error rate performance at high average SNR.

However, most previous research works (e.g., [28], [46], [49], [69], [70] and [74]) on transmission power allocation have failed to show the impact of the chosen optimization criterion, the performance gain achieved and the power allocation algorithm robustness. Targeting these limitations, Kostina developed an analytical tool in [32] to accomplish the optimum power allocation and compared various performance aspects based on various

optimization criteria; i.e., average optimization based on average total bit error rate (TBER) or block error rate (BLER) and instantaneous optimization based on instantaneous TBER/BLER. However, the analysis was limited to a specific constellation, Binary Phase Shift Keying (BPSK). The derivations in the aforementioned research work are also limited to specific constellations. In this thesis, the work in [32] is extended from BPSK to high-level modulation schemes whose constellations are suitable for adaptive modulation employed by practical systems. The insightful approach to quantify the power allocations and associated performance gain in [32] has also been extended into a unified framework for various high-level modulation techniques. Note that the unified framework in this thesis is investigated only for the power-optimized V-BLAST based on the average BLER to avoid the error propagation analysis.

In particular, the above extensions are based on a modification of the instantaneous Symbol Error Rate (SER) union bound concept. This modification, an “average union bound (AUB) of SER,” which is based on the minimum Euclidean distance in the constellation, not only provides a compact, closed-form SER upper bound, but also offers an insight into how the minimum distance affects the system performance. Although few studies proposed similar approximations for the higher-order constellations over additive white Gaussian noise (AWGN) channel ([15] and [44]), this thesis demonstrates that the approximation can be used in not only single-input single-output (SISO) Rayleigh but also MIMO channels to get a close bound of average SER for the whole range of SNR. In fact, this approximation of the instantaneous SER for different modulations is consistently substituted into the average SER expression of BPSK over diversity fading channels. Thus, the well-known closed form average SER expression for BPSK with maximum ratio combining (MRC) ([50]) has been generalized as an AUB of the average SER

for various constellations. To illustrate the accuracy of this general bound, Monte Carlo (MC) simulations and exact performance expressions are employed for comparison purposes.

Ultimately, the validity of our general optimized power allocation framework is verified with MC simulations, where random symbols are generated independently and the error events are determined by comparing the symbols to the output of the demodulator. As has been noted in [27] and [65], the MC estimation accuracy, which is indicated by the variance of the estimator, is proportional to the number of simulation trials. To this end, because the order of SER in this thesis is $10^{-4} \sim 10^{-6}$, the number of MC trails with 90% confidence needs to be up to 10^9 ([27] and [65]). Since the simulations with this amount of trails are computationally demanding, we study how to speed up MC simulations by using parallel Matlab® on a multi-processor computer without penalty in the estimation accuracy. During the study, the existing parallelism schemes and toolkits are reviewed and evaluated. Consequently, original Matlab® codes for V-BLAST simulation are “parallelized” by using the toolkits “pMatlab”. Finally, interesting observations regarding parallelism efficiency with respect of the run-time performance are summarized.

1.2. Contributions of the Thesis

The major contributions of the thesis are as follows:

- A unified closed-form approximation, AUB of the average SER over multi-path Rayleigh fading channels for different modulations is developed by using two constellation-related parameters; i.e., the normalized minimum pair-wise Euclidean distance and the average number of neighbors with this smallest Euclidean distance (sections 4.2 and 4.3); Exact SER expressions and MC simulations are used to verify the validity of AUB.

- The power-optimized ZF-VBLAST solution based on the average BLER for BPSK in [32] is extended to high-level modulations by using the above AUB model; The two constellation-related parameters reveal how the constellation properties affect the optimum solutions (section 5.2);
- The behavior of optimum power allocation coefficients at large and small average SNR is analyzed; The difference between the coefficients corresponding to lower-level and high-level constellations is also discussed (section 5.2);
- Monte Carlo (MC) simulations imitating power-optimized ZF-VBLAST for different modulations are implemented with parallel Matlab®; (i.e., we pick QPSK and 8-QAM for imitating high-level modulations with reasonable simulation time. Note that the implementations for other higher order constellations are same as the implementations for QPSK/8QAM and the performance variations when the constellation size increases are demonstrated in these two examples as well.) The utility of non-uniform power allocation is demonstrated and the simulation efficiency is doubled on a dual-core computer (i.e., the run time is half-cut) (section 5.2);
- The comparative studies of various existing parallel Matlab® schemes and toolkits are conducted and the cautions regarding applying parallel Matlab® toolkits (i.e., “pMatlab,” from Massachusetts Institute of Technology (MIT), and the Parallel Computing Toolbox™ (PCT), from Mathworks™) are pointed out (Chapter 8);
- The BLER performance of the power-optimized ZF-VBLAST is evaluated numerically and analytically; The performance behavior for different constellations is investigated (sections 5.2 and 7.1);

- The “local” and “global” robustness of the optimum power allocation algorithm are rigorously defined; The impact of minimum Euclidean distance on the robustness measure is demonstrated (sections 6.1 and 6.2);
- The SNR gain of the proposed optimization, as well as the behavior of this gain at large and small average SNR is rigorously defined and quantified (section 7.2 and 7.3); and the dependence of the SNR gain on the constellation size is examined.

1.3. Organization of the Thesis

This thesis is organized as follows. Chapter 2 reviews the history of research on MIMO and interprets a few important concepts in this area. BLAST algorithms are elaborated upon and followed by a brief review of the recent work on V-BLAST simplifications. Chapter 3 discusses zero-forcing (ZF), Maximum SNR and Minimum Mean-Square Error (MMSE) VBLAST algorithms and proposes a closed-form expression of the average BLER, which is the target function of the optimizations in this thesis. Chapter 4 proposes an approach to extend the original SER expression of the diversity-combining BPSK to a unified SER expression (i.e., the AUB of SER) of various high-order constellations. Numerical simulations of MPSK and MQAM corroborate the validity of this generalization. The rest of this thesis deals with the performance analysis of this unified power-optimized V-BLAST algorithm. Chapter 5 applies the above generalization in deriving the average BLER-based optimum power allocation for ZF-VBLAST. MC simulation results show the performance improvement achieved by this optimization. Chapter 6 examines the robustness measure of this optimization for a variety of modulation formats. Chapter 7 studies the SNR gain attained by using this optimization. Chapter 8 screens existing parallel calculation schemes and parallel Matlab® toolkits and introduces a few

important tips for applying parallel Matlab® in realistic simulations of error rates. Chapter 9 concludes the thesis.

Chapter 2. Literature Review

In the last decade, the demands of high data rate wireless links in internet-intensive networks have emerged as a challenge for communication research. In the same way that code-division-multiple-access (CDMA) technology made a marked breakthrough in the early 1990s, the new MIMO space-time coding (STC) is considered one of the most promising technologies for moving the industry forward. In fact, quite a few MIMO techniques and algorithms have been adopted by cutting-edge commercial standards and applied to the associated equipment; e.g., IEEE 802.11 for wireless local area networks (WLAN), ITU IMT-2000 for universal mobile telecommunication systems (UMTS), and associated wireless routers or base station products.

Generally, all the innovations advancing wireless channel improvements aim to achieve a compromise between capacity (the data rate) gain and performance (bit error rate (BER)) degradation. On this point, MIMO/STC is analogous to CDMA—the signal streams from single or multiple users sharing the same carrier frequency are recovered individually from the received signal mix at minimum cost to BER. Here, the difference between these two technologies is the domain being “exploited” to increase the capacity: the “code” domain for CDMA and the “space” domain for MIMO. Accordingly, CDMA compares the streams (associated with multiple users) to predefined “quasi-orthogonal” codes with an acceptable cost of extra frequency spreading, while the MIMO system utilizes the close-to-orthogonal nature of wireless channels in a rich multi-path environment without additional spectrum cost. The increased capacity advantage in MIMO/STC systems, called “spatial multiplexing,” is intuitively simple—the signal from the user is demultiplexed to multiple streams and these streams are transmitted through spatially distributed antennas. However, recovering these streams (associated with a single user) individually at the receiver is more complicated. The channel matrix makes all

MIMO research—e.g., channel modeling, channel capacity and signal processing—different from the traditional work. Besides “spatial multiplexing,” “spatial diversity” which is the diversity order introduced by uncorrelated spatial streams, is another important concept in MIMO communication. Exploiting this spatial diversity is done mainly by relying on space-time coding at the transmitter and diversity combining and successive interference cancellation (SIC) at the receiver. (Fig 2.1)

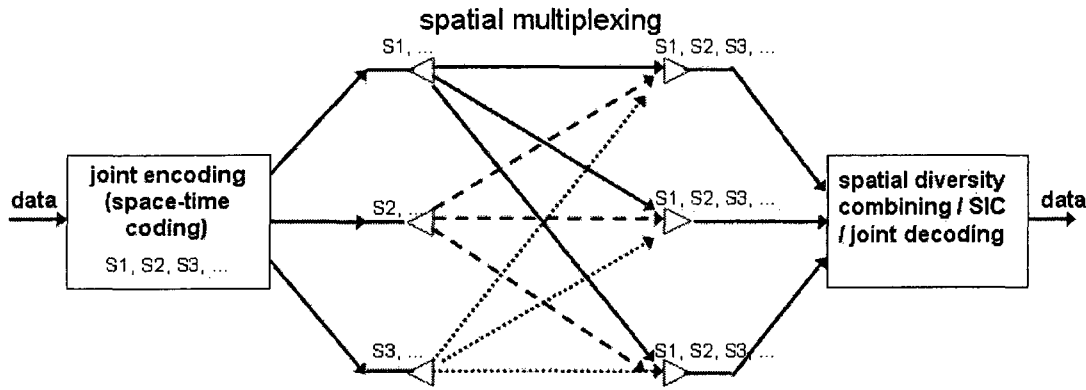


Fig. 2.1 MIMO architecture

Because it is the computationally demanding SIC step in the original V-BLAST that achieves spatial multiplexing and still provides some spatial diversity, we tried to replace SIC with a computationally efficient alternative in this thesis. Before we examine the performance of the V-BLAST algorithm in the next chapter, we briefly review current MIMO/STC research works in the following sections: 1) in the “spatial multiplexing” section: the major investigations of MIMO channel modeling and the fundamental promises of the MIMO capacity from the information theory perspective; 2) in the “spatial diversity” section: the basic ideas and advancements in space-time coding (achieving transmit diversity in the case of flat/frequency selective fading, with or without prior channel knowledge) and the signal processing constituents at the receiver; and 3) in the second part of the “spatial diversity” section: the examples with

both spatial multiplexing and spatial diversity—Diagonal Bell Laboratories Layered Space-Time (D-BLAST) and V-BLAST—with a variety of recent studies on the alternatives to V-BLAST.

2.1. Spatial Multiplexing

The MIMO/STC system is appealing because spatial multiplexing is proven to linearly multiply the system capacity by $\min(n, m)$; i.e., the number of uncorrelated transmitter-receiver antenna pairs given m transmitter antenna branches and n receiver antenna branches in a full scattering environment. Notably, the superiority of MIMO over single-input and multi-output (SIMO) and multi-input and single-output (MISO), where the capacity increases in a logarithmic fashion with m or n , is achieved at no extra spectrum cost (with only the penalty of hardware complexity). This MIMO capacity gain, which is proportional to $\min(n, m)$, was first discovered by [71] and analytically derived by [20] and [64] from two different viewpoints. As can be seen in the famous capacity equation given in [20] and [64], the MIMO capacity, a function of the $m \times n$ random channel matrix $\mathbf{H}$, is closely related to the statistic properties of the channel matrix $\mathbf{H}$. Consequently, MIMO channel modeling, parameterized by the antenna correlations and the eigenvalues of $\mathbf{H}$, is critical for assessing the MIMO spatial multiplexing gain.

1. MIMO Channel Capacity

The reason that the MIMO capacity is proportional to the number of transmitter-receiver pairs is that the coherent detection at the receiver can extract and combine the transmitted signal streams independently, thus the wireless channel with $(1/\min(n, m))$ bps/Hz bandwidth can actually carry 1 bps/Hz data stream without spending any additional power; e.g., with V-BLAST, [72] demonstrates a 20~40 bps/Hz spectrum efficiency in an indoor environment. An algebraic

analogy made by [23] is also provides an intuitive understanding—the m signal streams in a MIMO system are similar to the m unknowns resolved from the m independent equations. The transmitted streams are reminiscent of the unknowns and the rank of the channel matrix is equivalent to the number of independent equations. Of course, here the rank of $\mathbf{H}$ is equal to or less than $\min(n, m)$. (For cases where the rank is less than $\min(n, m)$, the possible reasons are the spatial correlation between the transmitter/receiver antenna branches, LOS components in the propagation, etc. ([9]))

Currently, as the fundamental capacity in [20] and [64] is only achieved under ideal conditions—where the transmission power is equally allocated among the uncorrelated data streams and the channel is made up of i.i.d. Rayleigh flat fading multipath—advanced research into information theoretic MIMO capacity focuses on examining the influence of the channel/data statistic characteristics on the capacity in real wireless environments. First, the statistic properties of the capacity, as a random variable dependent on $\mathbf{H}$, are revealed right after Foschini and Teltar’s famous observation ([17], [29], [30], [51], [64] and [55]). Later, the work can be categorized as maximizing the capacity when the channel is known/unknown at the transmitter. When $\mathbf{H}$ is known at the Tx, the capacity is optimized over the covariance of data streams by [18] and [67], where the optimum solution is termed “waterfilling” (WF). In a scenario with unknown $\mathbf{H}$ at the transmitter, the impact of data stream correlations on capacity is indicated analytically in [56]. In a similar unknown scenario, a specific case—MIMO system capacity over a Ricean channel—was studied by [17] and [18]. As has been noted, the analytical derivations in the abovementioned literature are complicated; recent research has shown interest in giving asymptotical capacity bounds for different spatially correlated or interference corrupted

MIMO channels ([29], [36] and [45]). Furthermore, a promising current study has modeled the capacity approximations for real cellular systems ([54] and [73]).

2. MIMO Channel Modeling

Considering that the propagation conditions in the real world are important to MIMO capacity, we may conclude that channel modeling is crucial for capacity analysis. In fact, all the conclusions or analysis in the literature described above are derived under certain channel models. Thus, the goal is to introduce spatial dimensions into those existing propagation models of reflection, diffraction and scattering. [7], [25] and [47] have provided a few good models of the MIMO channel and its corresponding statistical properties. Accordingly, [22] and [23] summarize that the key modeling factors include joint antenna correlations and distribution of eigenvalues in the channel matrix $\mathbf{H} \mathbf{H}^+$. ($\mathbf{H}^+$ is the Hermitian transpose of $\mathbf{H}$.) For example, [22] splits $\mathbf{H}$ into line of sight (LOS) and non-line of sight (NLOS) components and demonstrates that the LOS components lower the rank of $\mathbf{H}$ and the degrees of freedom of spatial multiplexing, hence decreasing MIMO capacity. However, the joint antenna correlation quantified by the correlation matrix of NLOS components (i.e., a Kronecker product of the covariance matrices at transmitter and receiver) is indeed subjected to the power azimuth distribution (PAD) ([22] and [56]). Moreover, for the indoor environment, as scattering is rich, [60] has shown the LOS portion is diminished and the NLOS portion is approximately Rayleigh distributed. To this end, [66] further analytically proves that i.i.d. Rayleigh fading is a reliable model for this rich-scattering environment, given widely-spaced antennas.

After all, the industry defined two important models (i.e., [1] and [77]) for 3GPP and IEEE802.16 separately in 2001). Also the “pinhole” or “keyhole” effect has been the subject of

recent studies on MIMO implementations in an outdoor environment. In this case, the scattering closest to the transceiver leads to a correlation decrease and a low $\mathbf{H}$ rank ([10] and [22]). Accordingly, [10] first demonstrated the occurrence of “keyhole” effect with a physical example and proposed that the rank-deficient channel leads to capacity degradation. [22] examined the effect of the keyhole on ergodic capacity by using a parameterized scattering MIMO model. Recently, in [37], an asymptotic analysis of the single-keyhole channels’ impact on the outage capacity and instantaneous SNR was introduced. Above all, [66] is a good reference, dwelling on the physical properties mentioned above, e.g., angle-of-arrival (AOA), antenna spacing and the geographical separation of paths and capturing their effects in succinct statistic channel models.

2.2. Spatial Diversity

The basic idea of spatial diversity is that when multiple wireless data streams are transmitted over independent paths, the probability that all these streams are simultaneously in deep fading is extremely small. Hence, spatial diversity succeeds in improving the “soft” transmission outage capacity, instead of the “hard” average throughput capacity increased by spatial multiplexing. Given the scarcity of the radio spectrum, the transmitter power and the receiver sensitivity, the spatial diversity is a new promising “dimension” to capacity improvements ([48]).

In fact, before Telatar’s and Foschini’s seminal papers, the research on spatial diversity exploitation focused on dealing with diversity reception (SIMO) and transmission diversity (MISO) separately. The idea of SIMO is conceivable here: the receiver may independently combine diverse copies of the same signal coherently to improve performance, whereas the transmit diversity utilized in MISO is sophisticated: to protect the original data, it should be

divided into “streams,” and the individual streams are jointly encoded. This mutual redundancy minimizes the outage probability and consequently maximizes the outage capacity. Unlike the previous systematic studies on diversity reception (e.g., the receiver diversity adopted by Global System for Mobile (GSM) communications, in which two receiver antennas are configured on a typical base station), the research work on transmit diversity was unremarkable in the early 1990s, until the breakthrough achieved by [2] and [62].

It was only after Telatar and Foschini’s work that researchers realized the importance of synthesizing transmit and receive spatial diversities, especially for MIMO systems. Today, although both space-time coding (STC) at the transmitter and successive interference cancelling (SIC) at the receiver are widely conceived as indispensable, the practical application problems still remain: the tradeoff between diversity gain and the implementation difficulty of the diversity extraction; i.e., when more data streams are employed, more complex diversity combining and space-time coding schemes are expected.

3. *Space-Time Coding*

Although the information theoretic capacity upper bound in [64] and [20] can be achieved using trellis codes and Viterbi decoding ([59]), the associated complexity and latency penalty are inappropriate for a real system. Therefore, despite the significant performance boost achieved by space time trellis codes (STTC) ([62]), the space time block codes (STBC) associated with favorable linear decoding at no cost of diversity gain inspired more research activity after it was first proposed ([2] and [23]). Moreover, STBC has also proved instrumental in improving the performance of current commercial systems, from enhanced data rates for GSM evolution (EDGE) to IEEE 802.11 ([14], [41] and [40]). Consequently, Alamouti code became the

constituent code of Universal Mobile Telecommunications System (UMTS) standard in 2002 due to its simple conception ([16]).

Despite its strong influence, described above, there are two limitations in Alamouti's work: 1) it loses additional SNR gain offered by trellis coding; i.e., in the STBC scheme, the original symbols are sent with their orthogonal redundant copies among the antenna branches in a criss-cross pattern "block" by "block", where only minimal coding gain is provided; 2) the derivation was limited to the real constellations with two transmission antennas. For better transmission performance, some of the later literature proposed concatenating the STBC and the outer trellis channel coding scheme (STBC-TCM) ([3], [5], [24] and [58]), but there is associated transmission rate loss. With regard to the second issue, [63] further extends the original STBC to a general orthogonal STBC designed for arbitrary constellations (e.g., QAM and MPSK) and transmit antenna numbers, but only half to three-quarters of the maximum possible rate is achieved. Correspondingly, [39] finally proves that the general "rate of one" STBC doesn't exist in the abovementioned arbitrary constellation and transmit antennas scenarios. Summarizing most of the above work, two classic reference books, [48] and [34], discuss different variants of ST coding schemes. Most significantly, [34] further states that the core idea in constructing a specific STBC for any given constellation and transceiver antennas is finding an encoding matrix. This matrix, as a linear combination of the real and imaginary parts of the symbol alphabet, can be obtained by various criteria, such as simplifying maximum likelihood detection, minimizing error probability and maximizing data throughput.

Notably, in most of the references mentioned above, the channel is assumed to be flat fading, while for frequency selective MIMO channels plus bandwidth-delay spread, orthogonal frequency-division multiplexing (OFDM) uses a cyclic prefix (CP) to eliminate inter-symbol

interference (ISI). Virtues of OFDM include, but are not limited to: 1) by means of CP, all signal processing in the equalization can be implemented with fast Fourier transforms (FFT); and 2) the FFT of the data streams at the transceiver causes the frequency-selective fading channel to act as uncorrelated parallel flat channels ([34]). Recently, some literature has dealt with the combination of space frequency block-coding (SFBC) and OFDM, where the proposed system exploits the frequency diversities and outperforms the traditional STBC, especially [26], [38], [35], [31], and [75]. However, how to combine CP with the SFBC/STBC encoding scheme efficiently and apply select combining and equalizations with low complexity remain open problems for research on WLAN or 4G systems.

4. *Successive Interference Cancellation—BLAST*

Although [20] and [62] conceived the information-theoretic capacity in a MIMO channel, how to carry out both spatial multiplexing and spatial diversity (i.e., recover individual signal streams from the received signal mix) is not clear. The maximum likelihood (ML) receiver provides the best performance, but is subject to a complexity penalty as well. Hence, linear detectors, which are less complex than ML detectors, are more desirable for MIMO detection. But the performances of either zero-forcing (ZF) or minimum mean-squared error (MMSE) detectors are by no means satisfactory ([4]). Consequently, decision feedback detection, also called successive interference cancellation (SIC), was originally employed by Wolniansky and Foschini in [19] and termed Bell Laboratories Layered Space-Time (BLAST)¹. It is a computationally effective signal processing architecture and yet achieves a sizable portion of the full MIMO capacity.

¹ The term “Vertical Bell Laboratories Layered Space-Time” (V-BLAST) was proposed in 1998 by Wolniansky and Foschini in [Wolniansky, 1998]. However, the exact ideas were first proposed by Foschini in [Foschini, 1996], where both architectures of V-BLAST and D-BLAST were investigated.

In [19], two BLAST schemes, vertical BLAST (V-BLAST) and diagonal BLAST (D-BLAST), are developed. The SIC steps in V-BLAST are shown in Figure 2.2. When these steps are compared to the procedures in a decision-feedback equalizer (DFE), the similarities are clear: the “interference cancellation” works as the feedback filter, canceling the interference from the previous estimated symbols, and the “interference nulling” diminishes the interference from the subsequent received symbols, in the same way that the feed-forward filter does. For D-BLAST, on the other hand, the only difference is the additional vector encoding process. Unlike V-BLAST, in which each data stream associated with a certain transmitter antenna is laid “vertically,” D-BLAST jointly encodes data streams and extends each codeword “diagonally” to the transmitter antennas. At the receiver, the codeword is decoded only after all “segments” of the same codeword have been obtained from the receiver antennas. Obviously, the key idea here is to improve system outage performance by averaging the randomness in individual channels. Details on D-BLAST are beyond the scope of this thesis and can be found in the cited literature.

V-BLAST opens an appealing horizon for researchers, as it provides a balance between detection complexity and achievable capacity/diversity. Although V-BLAST achieves significant portions of the full MIMO capacity and diversity order ([21][42]), implementing the traditional V-BLAST in a realistic operating scenario is still demanding. The major obstacle is the high computationally intensive ordering procedure in each layer peeling step. Because the error propagation in SIC makes this optimal ordering indispensable, other computationally effective implementations or alternatives to sequential ordering should be adopted. Therefore, researchers have developed various modifications, which allow them to: 1) simplify the conventional repeated optimal ordering, as well as the associated interference nulling vectors generation; 2) develop alternatives to the traditional optimal ordering for error propagation alleviation.

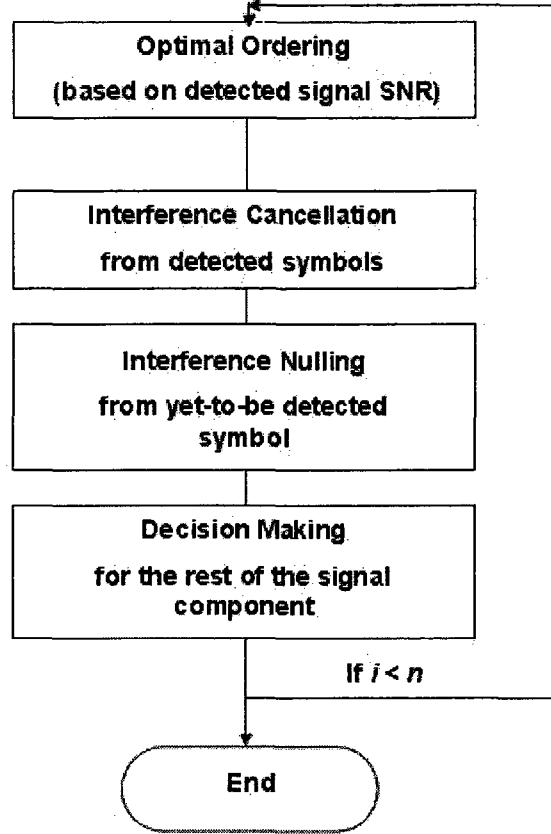


Fig.2.2 V-BLAST detection procedures

Of those efforts targeting the first issue, [33] is heuristic. Targeting the computational cost, Hassibi developed a so-called square root algorithm, in which the ordered nulling vectors in each SIC step are not derived from the sequential pseudoinverse of the deflated channel matrix $\mathbf{H}_{i-1}$ (see the following chapter for details on nulling vector derivation in the original V-BLAST algorithm), but from an orthogonal transformation based on a one-time QR decomposition of $\mathbf{H}$ instead. Moreover, [33] also numerically analyzes the computational complexity associated with the traditional V-BLAST and demonstrates that the proposed square root solution may offer an order of magnitude of savings in the computational cost when the number of antennas is large. Further, the numerical robustness in Hassibi's algorithm is also improved by replacing squaring and inverting operations with unitary transformation of the target matrix. After Hassibi's work,

[76] provides a modified correlating decision feedback detection, further reducing the complexity of the square root algorithm. With this similar decision feedback (DF) algorithm as the decorrelating multi-user detection in CDMA, Zha et al. showed that optimal ordering can be derived succinctly via cross-correlation matrix of $\mathbf{H}$ and its Cholesky decomposition. For $m = n$ cases, [76] shows that its computational complexity is only 26% of that in the square root algorithm. Alternatively, with the help of recursive computation based on the Sherman-Morrison formula, [6] suggests that optimal ordering can also be derived without any pseudoinverse operation of $\mathbf{H}$. Note that this recursive fast detection method is practical irrespective of the numerical instabilities introduced by recursion, because the number of receiver antennas n , which is also the number of the recursions, is limited in a real MIMO system.

Even though the above algorithms simplify original successive ordering with various low computationally complexity “one-time determinations,” those associated with QR decomposition, Cholesky decomposition and recursion are still intensive. Note also that the bottom line is to develop methods to compensate for error rate performance degradation caused by optimal ordering exclusion. Taking account of the dominating errors caused by the first peeling step, [11] realized the above basic idea in an intuitively straightforward way: introducing maximum likelihood (ML) detection in the first two iteration steps, and the related decision is fed back to the following detection steps. The diversity of the first (i.e., the worst) “peeled” signal stream is then improved by the ML decoding, and the decision feedback equalizer (DFE) then extracts the decent decoding performance of the other streams. Later on, Choi suggested a novel iterative soft interference cancellation scheme in [12], where the tentative soft decisions and corresponding posterior probabilities of all possible transmitted symbols are made to advance the detection. Additionally, [12] numerically exemplifies the proposed algorithm, which outperforms the

traditional unordered zero forcing (ZF) V-BLAST by 3~5dB SNR gain in a QAM system over a 10×10 MIMO system. Similar ideas are also suggested in [53], where the low diversity stream is decoded iteratively using the feedback decisions from the higher steps. Although the abovementioned papers do demonstrate a few performance improvements over the unordered V-BLAST, the associated decision feedback overhead and ML detection is still unfavorable for a real-time system.

Hence, constraining the joint transmit-receive optimizations described above to a non-uniform power allocation among the transmitter antennas in an unordered V-BLAST is regarded as feasible for a real-time system. In non-uniform power allocation, more power is allocated to the lower diversity stream to decrease dominating decoding errors, and hence, overall system error rate—which is mainly limited by the error propagation from the worse streams (i.e., lower diversity streams)—can be improved significantly with reasonable computational complexity. Note that the above “overall system error rate” could be TBER or BLER. TBER corresponds to the total bit errors over the merged detected streams and BLER quantifies the probability of having at least one symbol error in the detection output vector. Furthermore, the optimum coefficients can be derived with instantaneous error rate or average error rate-based criteria. The instantaneous power allocation techniques deal with the instantaneous error rates corresponding to every single channel realization, and the average power allocation bases its adjustment on the average error rate over all the channel realizations.

In the literature dealing with optimum power allocations, [46], [69] and [70] uses instantaneous TBER as the optimization criterion. All of these publications present closed-form optimum power allocations for ZF or MMSE-VBLAST with arbitrary transmitter and receiver antennas. In [69], error propagation is neglected, but solutions are given for V-BLAST both with

and without ordering. In [70], the authors further elaborate on their optimization solutions for both ZF and MMSE-VBLAST. Error propagation is analyzed in [46], and the average BER is minimized for every transmitted symbol. Optimization based on asymptotic TBER approximation offers 4dB SNR gain in a 4×4 ZF-VBLAST compared to the traditional ZF-VBLAST at a TBER of 10^{-3} . Referring to the other optimization criterion-instantaneous BLER, [28] achieves 2dB SNR gain in an optimized $2 \times n$ ordered V-BLAST at BER of 10^{-3} . Obviously, the schemes obtained by the author of [46] and [70] are limited by the intractable exact analysis of TBER, due to the error propagation effect. In fact, as a result of low error propagation probability between decoding steps, the difference in the power allocation solutions based on instantaneous TBER and BLER is very small ([32]). Moreover, a less complex optimization approach can be adopted by using the average rather than instantaneous error rate criterion ([49] and [74]). In a real system suffering from small-scale fading, occasional channel information feedback and power re-allocations are preferable to continuous real-time feedback and adjustment.

In addition to the numerical solutions given by [74] and [49], [32] further develops the average TBLER and BLER-based power allocation solutions analytically, using an insightful approach. Moreover, [32] makes the performance comparison of the instantaneous and the average optimizations. Additionally, [32] presents about various performance aspects (i.e., the algorithm robustness and achieved SNR gain) of instantaneous and average optimizations as well.

However, the limitation of the work in [32] is that only a specific modulation, BPSK, is analyzed. As a general explicit power allocation for high-level constellations is indispensable to a realistic MIMO implementation, this thesis adopts the approaches provided by [32], but also extends the derivation to high-level modulations. Significantly, the generalization is obtained by

using a generic approximate BLER expression, and this expression provides new insights into which factors affect the BLER performance of an unordered V-BLAST, and how. The major optimization algorithm performance measures of [32], such as algorithm robustness and SNR gain achieved, are also studied in this thesis to corroborate the validity of the proposed universal optimization framework.

2.3. Summary

Spatial multiplexing and spatial diversity combining, the two fundamental constituent parts of MIMO technology, were briefly discussed in this chapter. Further, this chapter sketched the advancements achieved by the research, which can be categorized as MIMO capacity, MIMO channel modeling, space-time coding and transceiver signal processing. It is apparent that all previous work has aimed to achieve the theoretical capacity predicted by Telatar and Foschini in a compact signal processing architecture. Among the proposed architectures, V-BLAST of Foschini et al. attains a reasonable complexity-performance balance in a rich-scattering environment. Moreover, a variety of alternatives have been developed to simplify the complicated sequential optimal ordering in the original V-BLAST algorithm, which relies on a computationally complicated pseudoinverse operation of the channel matrix. Above all, non-uniform power allocation among transmitter antennas is regarded as an attractive solution among these optimal ordering alternatives. However, most studies on this arrangement did not indicate analytical derivations of power allocations and the associated performance improvements, until [32] set up an analytical derivation framework for optimizations and performance evaluations. This thesis extends Kostina's efforts from BPSK to general higher-level constellations by

introducing a generic BLER approximation. Consequently, this thesis describes a few interesting performance observations related to the constellation size.

Chapter 3. V-BLAST Algorithm and its Performance

In this chapter, various V-BLAST algorithms and the performance of ZF-VBLAST proposed/investigated in [20], [72], [42] and [43] are reviewed, because they are essential to the analysis in the following chapters.

Before the introduction of V-BLAST performance, the assumptions in this thesis should be clarified:

- Flat-fading quasi-static channel; i.e., the channel gain is time-varying from symbol to symbol, but stable during the symbol interval, with Additive White Gaussian Noise (AWGN).
- The transmitted signal streams are independent as are the channel gains.
- The receiver has perfect knowledge of the channel through a training sequence.
- The synchronization is perfect.

The standard representations of the baseband discrete MIMO system in this thesis include the following: 1) m independent signal “streams” are transmitted through m transmitter antennas; and 2) these m parallel streams are received by n antennas, where symbol streams are mixed. The transmitted signal vector $\mathbf{s}_{m \times 1}$ and the received vector $\mathbf{r}_{n \times 1}$ are related as

$$\mathbf{r} = \mathbf{H} \cdot \mathbf{s} + \boldsymbol{\xi} = \sum_{i=1}^m \mathbf{h}_i \mathbf{s}_i + \boldsymbol{\xi} \quad (3.1)$$

where $\boldsymbol{\xi}_{n \times 1}$ is the AWGN vector with independent Gaussian random elements $\xi_i \sim \mathcal{CN}(\mathbf{0}, \sigma_0^2 \mathbf{I})$.

$\mathbf{H}$ is the $n \times m$ channel gain matrix with elements h_{ij} representing the gain between i th receive and j th transmit antennas.

3.1. V-BLAST Algorithm

As mentioned in Chapter 2, V-BLAST is a suboptimum SIC algorithm, which is feasible because of its good balance between performance and complexity. With regard to the different choices of the nulling weighting vectors in the interference nulling procedure, V-BLAST can be categorized as Zero Forcing (ZF) and Maximum SNR and Minimum Mean Square Error (MMSE) V-BLAST. Although the derivations in this thesis are based on the simplest algorithm, ZF-VBLAST, the other two V-BLAST algorithms are discussed as well.

Optimal Ordering: In the first layer peeling step of V-BLAST, the correlations between the channel vector $\mathbf{h}_k$, $k = (1, \dots, m)$ and the subspace with the span $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}, \mathbf{h}_{k+1}, \dots, \mathbf{h}_m\}$ are examined to find the largest measure of

$$a_{1,k} = \left| \mathbf{C}_1^{(k)} \mathbf{h}_k \right| \quad (3.2)$$

where $\mathbf{C}_1^{(k)}$ is a projection matrix to the subspace orthogonal to $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}, \mathbf{h}_{k+1}, \dots, \mathbf{h}_m\}$.

$\mathbf{C}_1^{(k)}$ is given by²

$$\mathbf{C}_1^{(k)} = \mathbf{I} - \mathbf{H}_0^{(k)} \left(\mathbf{H}_0^{(k)} \mathbf{H}_0^{(k)+} \right)^{-1} \mathbf{H}_0^{(k)+} \quad (3.3)$$

where $\mathbf{H}_0^{(k)} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}, \mathbf{h}_{k+1}, \dots, \mathbf{h}_m]$ and the stream $j_0 \in [1, \dots, m]$ is then selected:

$$j_0 = \arg \max_k \left| \mathbf{C}_1^{(k)} \mathbf{h}_k \right| \quad (3.4)$$

In the second layer peeling step, if $\mathbf{H}_1 = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{j_0-1}, \mathbf{h}_{j_0+1}, \dots, \mathbf{h}_m]$ denotes the space with the span of remaining channel vectors, the optimal ordering then calculates the correlations between the channel gain vector $\mathbf{h}_k$, $k = (1, \dots, j_0 - 1, j_0 + 1, \dots, m - 1)$ and the subspace with the span $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}, \mathbf{h}_{k+1}, \dots, \mathbf{h}_{j_0-1}, \mathbf{h}_{j_0+1}, \dots, \mathbf{h}_m\}$. The stream j_1 is selected as

² As the pseudoinverse of $\mathbf{H}^{(k)}$ is $(\mathbf{H}^{(k)} \mathbf{H}^{(k)+})^{-1} \mathbf{H}^{(k)+}$ and $\mathbf{H}^{(k)} (\mathbf{H}^{(k)} \mathbf{H}^{(k)+})^{-1} \mathbf{H}^{(k)+}$ is the projector on the span of $\mathbf{H}^{(k)}$, $\mathbf{I} - \mathbf{H}^{(k)} (\mathbf{H}^{(k)} \mathbf{H}^{(k)+})^{-1} \mathbf{H}^{(k)+}$ then turns to the projector on the subspace orthogonal to $\mathbf{H}^{(k)}$.

$$j_1 = \arg \max_k \left| \mathbf{C}_2^{(k)} \mathbf{h}_k \right|$$

where

$$\mathbf{C}_2^{(k)} = \mathbf{I} - \mathbf{H}_1^{(k)} \left(\mathbf{H}_1^{(k)} \mathbf{H}_1^{(k)+} \right)^{-1} \mathbf{H}_1^{(k)+}$$

and $\mathbf{H}_1^{(k)} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}, \mathbf{h}_{k+1}, \dots, \mathbf{h}_{j_0-1}, \mathbf{h}_{j_0+1}, \dots, \mathbf{h}_m]$. Generally, the above operations are repeated in each step i , but the target correlation measure and subspace are $a_{i,k} = \left| \mathbf{C}_i^{(k)} \mathbf{h}_k \right|$ and $\mathbf{H}_{i-1}^{(k)}$ instead.

The idea here is to prioritize the streams and hand the lower SNR streams to later steps; i.e., those $\mathbf{h}_k$ that are highly correlated with the remaining $(m - i - 1)$ channel vectors are left for later steps. (See the following sections for explanations.) The new channel vectors after the optimal orderings will be denoted as $\mathbf{h}_k'$ and the signal vectors will be $\mathbf{s}_k'$. The (3.1) can be rewritten as:

$$\mathbf{r} = \sum_{i=1}^m \mathbf{h}_i \mathbf{s}_i + \xi \xrightarrow{\text{Optimal Ordering}} \sum_{i=1}^m \mathbf{h}'_i \mathbf{s}'_i + \xi$$

Interference Cancelling: Secondly, in the interference cancellation of step i , we find that interference from previous $i-1$ symbols can be diminished by

$$\mathbf{r}' = \mathbf{r} - \sum_{k=1}^{i-1} \mathbf{h}'_k \hat{\mathbf{s}}'_k = \sum_{l=i}^m \mathbf{h}'_l \mathbf{s}'_l + \xi \quad (3.5)$$

where $\mathbf{r}$ is the original received signal mix and $\hat{\mathbf{s}}'_k$ are the previous detected symbols.

Interference Nulling: Consequently, the nulling vectors can be chosen to satisfy either ZF or Maximizing SNR / MMSE criteria. For the ZF criterion, $\mathbf{r}'$ is projected to a subspace orthogonal to the space $\{\mathbf{h}'_{i+1}, \mathbf{h}'_{i+2}, \dots, \mathbf{h}'_m\}$

$$\mathbf{r}'' = \mathbf{P}_i \mathbf{r}' = \mathbf{P}_i \mathbf{H}_{(i-1)} \mathbf{s}'_{(i-1)} + \mathbf{P}_i \xi_i \quad (3.6)$$

where $\mathbf{H}_{(i-1)} = [\mathbf{h}'_i, \mathbf{h}'_{i+1}, \dots, \mathbf{h}'_m]$ and $\mathbf{s}'_{(i-1)} = [\mathbf{s}'_i, \mathbf{s}'_{i+1}, \dots, \mathbf{s}'_m]^T$. The inter-stream interference (ISI) can be eliminated by multiplication with $\mathbf{P}_i$:

$$\mathbf{P}_i \mathbf{H}_{(i-1)} \mathbf{s}'_{(i-1)} + \mathbf{P}_i \boldsymbol{\xi} = \mathbf{P}_i \mathbf{h}'_i s'_i + \mathbf{P}_i \boldsymbol{\xi} \quad (3.7)$$

where the projection matrix $\mathbf{P}_i$ is

$$\mathbf{P}_i = \mathbf{I} - \mathbf{H}_{(i)} \left(\mathbf{H}_{(i)} \mathbf{H}_{(i)}^+ \right)^{-1} \mathbf{H}_{(i)}^+ \quad (3.8)$$

given that $\mathbf{H}_{(i)} = [\mathbf{h}'_{i+1}, \dots, \mathbf{h}'_m]$. Maximum ratio combining (MRC) is used to improve the output SNR. The detection then becomes

$$\hat{s}'_i = \mathbf{w}_i^+ \mathbf{r}' = \mathbf{h}_i'^+ \mathbf{P}_i \mathbf{h}'_i s'_i + \mathbf{h}_i'^+ \mathbf{P}_i \boldsymbol{\xi}' \quad (3.9)$$

given the optimum weighting factor is $\mathbf{w}_i = \mathbf{P}_i \mathbf{h}'_i$. (See the following section for details.)

Ultimately, any ISI from the previous and subsequent symbols are cancelled, which is why it is called zero forcing.

For the Maximizing SNR criterion, the output SNR derived from (3.9) is maximized when

$$\begin{cases} \hat{s}'_i = \mathbf{w}_i^+ \mathbf{r}' = \mathbf{w}_i^+ \mathbf{h}'_i s'_i + \mathbf{w}_i^+ \boldsymbol{\xi}' = r_{si} + r_{\xi i} \\ r_{si} = \mathbf{w}_i^+ \mathbf{h}'_i s'_i \\ r_{\xi i} = \mathbf{w}_i^+ \boldsymbol{\xi}' = \sum_{k=i+1}^m \mathbf{w}_i^+ \mathbf{h}'_k s'_k + \mathbf{w}_i^+ \boldsymbol{\xi} \end{cases}$$

and

$$\begin{cases} P_s = \langle |r_{si}|^2 \rangle = |\mathbf{w}_i^+ \mathbf{h}'_i|^2 \\ P_\xi = \langle |r_{\xi i}|^2 \rangle = \langle \mathbf{w}_i^+ \mathbf{H}_{(i)} \mathbf{s}_{(i)} \mathbf{s}_{(i)}^+ \mathbf{H}_{(i)}^+ \mathbf{w}_i \rangle + \sigma_0^2 \mathbf{w}_i^+ \mathbf{w}_i \\ \quad = \mathbf{w}_i^+ (\sigma_0^2 \mathbf{I} + \mathbf{H}_{(i)} \mathbf{H}_{(i)}^+) \mathbf{w}_i \\ \gamma_i = \frac{P_s}{P_\xi} = \frac{\mathbf{w}_i^+ \mathbf{h}'_i \mathbf{h}_i'^+ \mathbf{w}_i}{\mathbf{w}_i^+ (\sigma_0^2 \mathbf{I} + \mathbf{H}_{(i)} \mathbf{H}_{(i)}^+) \mathbf{w}_i} \end{cases}$$

The solution is provided as (see Appendix A1 for the derivation):

$$\mathbf{w}_i = (\sigma_0^2 \mathbf{I} + \mathbf{H}_{(i)} \mathbf{H}_{(i)}^+)^{-1} \mathbf{h}'_i \quad (3.10)$$

Similarly, for the MMSE criterion, as the mean square error is

$$\varepsilon_i^2 = \left\langle \left| s'_i - \widehat{s}_i \right|^2 \right\rangle = \left\langle \left| s'_i - \mathbf{w}_i^+ \mathbf{r} \right|^2 \right\rangle$$

the optimum solution can be found by using $d\varepsilon_i^2/d\mathbf{w}_i = 0$. The weighting factor solution is the same as (3.10). (See Appendix A1 for the detailed derivation.)

3.2. Output SNR in ZF-VBLAST Steps

1. Distribution of Output SNR

Besides (3.8), there is another expression of the projection operation in the interference nulling ([42]):

$$\mathbf{P}_i = \frac{1}{|\mathbf{R}^{[i+1,m]}|} \begin{bmatrix} \boldsymbol{\eta}_i & \boldsymbol{\eta}_i \boldsymbol{\eta}_{i+1} & \cdots & \boldsymbol{\eta}_i \boldsymbol{\eta}_m \\ \boldsymbol{\eta}_{i+1} & & & \\ \vdots & & \mathbf{R}^{[i+1,m]} & \\ \boldsymbol{\eta}_m & & & \end{bmatrix} \quad (3.11)$$

where $\boldsymbol{\eta}_i = \mathbf{h}'_i / |\mathbf{h}'_i|$ is the normalized channel gain and $|\mathbf{h}'_i| = \sqrt{\sum_j |h_{ij}|^2}$. Here $\boldsymbol{\eta}_i \boldsymbol{\eta}_j = \sum_{k=1}^n \eta_{ki} \eta_{kj}^*$ is the inner product of $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_j$. $\mathbf{R}^{[i+1,m]}$, the “instantaneous channel correlation matrix” built on $[\boldsymbol{\eta}_{i+1}, \dots, \boldsymbol{\eta}_m]$, can be represented as

$$\mathbf{R}^{[i+1,m]} = \begin{bmatrix} 1 & \boldsymbol{\eta}_{i+1} \boldsymbol{\eta}_{i+2} & \cdots & \boldsymbol{\eta}_{i+1} \boldsymbol{\eta}_m \\ \boldsymbol{\eta}_{i+2} \boldsymbol{\eta}_{i+1} & 1 & \cdots & \boldsymbol{\eta}_{i+2} \boldsymbol{\eta}_m \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\eta}_m \boldsymbol{\eta}_{i+1} & \boldsymbol{\eta}_m \boldsymbol{\eta}_{i+2} & \cdots & 1 \end{bmatrix} \quad (3.12)$$

Therefore, the signal power in (3.7) is

$$|\mathbf{P}_i \mathbf{h}'_i s'_i|^2 = |s'_i|^2 |\mathbf{h}'_i|^2 \frac{|\mathbf{R}^{[i,m]}|}{|\mathbf{R}^{[i+1,m]}|} \quad (3.13)$$

On the other hand, as (3.6) shows, there is an n -dimension Gaussian noise vector at the first decoding step. Consequently, in each layer peeling step i , only the noise components orthogonal to the $(m - i)$ channel vectors remain. The dimensionality of this remaining Gaussian noise is $n - (m - i)$ ([42]).

$$\langle |\xi_i|^2 \rangle = (n - m + i) \sigma_0^2 \quad (3.14)$$

Considering the special case $i = 1$, we may intuitively expect that (3.13), which is the form of

$$|\mathbf{P}_1 \mathbf{h}'_1 s'_1|^2 = |s'_1|^2 |\mathbf{h}'_1|^2 (1 - |\boldsymbol{\eta}_1 \boldsymbol{\eta}_2|^2)$$

Furthermore, the analysis of the above output SNR should be extended to the higher decoding steps. Optimum ordering is skipped and the diversity order that the i th step achieved is demonstrated with the following geometrical representation of the nulling procedure ([42]). First, to simplify the analysis, a 2×2 case is considered, where $\mathbf{h}'_1$ and $\mathbf{h}'_2$ are represented as two-dimensional vectors. After interference nulling in the first step, the projection of $\mathbf{h}'_1$ is $\mathbf{h}'_{1\perp} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$, which is the component orthogonal to $\mathbf{h}'_2$. Figure 3.1 shows that $\mathbf{h}'_{1\perp}$ spans the dashed line—a one-dimensional sub-space; [42] further argues that if the subspace $[\mathbf{h}'_1, \mathbf{h}'_2]$ is rotated and $\mathbf{h}_2$ is parallel to one of the two basic axes, the $\mathbf{h}'_{1\perp}$ is a non-zero component in the other basic axis. Hence, the magnitude $|\mathbf{h}'_{1\perp}|^2 \sim \chi^2_2$ is two degree of freedom chi-square distributed. Note that the “two degree of freedom” refers to the two Gaussian variables comprising the chi-square random variable. Similarly, in a 2×3 case represented by three-dimensional space, as Figure 3.2 shows, $\mathbf{h}'_{1\perp}$ spans a two-dimensional sub-space orthogonal to $\mathbf{h}'_2$. If this three-dimensional space is rotated and one basic axis is parallel to $\mathbf{h}'_2$, $\mathbf{h}'_{1\perp}$ has two non-zero components. Thus, there is $|\mathbf{h}'_{1\perp}|^2 \sim \chi^2_{2 \times 2}$.

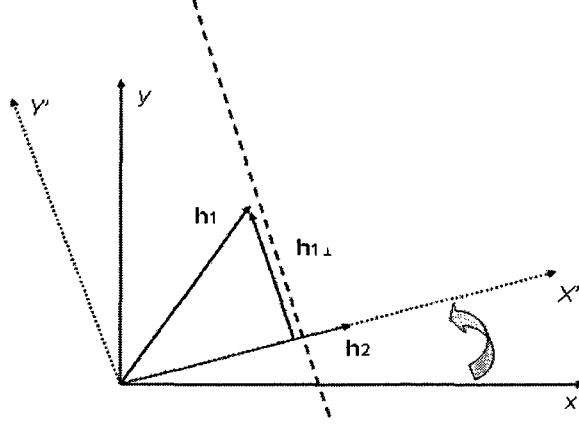


Fig. 3.1 Geometrical representation of the signal output after interference nulling (2D)

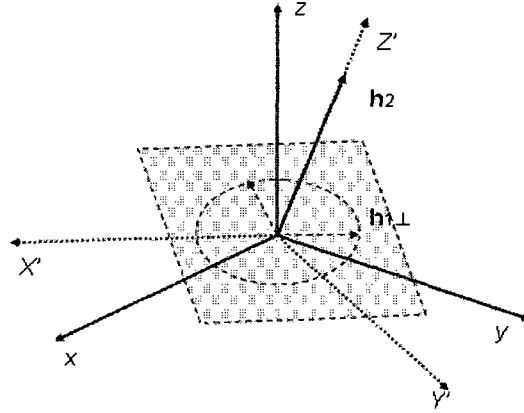


Fig. 3.2 Geometrical representation of signal output vector after interference nulling (3D)

Generally, although the representation of a $2 \times n$ detection is transcendental, it is conceivable that $\mathbf{h}'_{1\perp}$ spans an $n - 1$ dimensional subspace in the first step and the signal power after projection is $|\mathbf{h}'_{1\perp}|^2 \sim \chi^2_{2 \times (n-1)}$. In more general cases (i.e., $m \times n$ V-BLAST), the signal component $\mathbf{h}'_{i\perp}$ orthogonal to the remaining $(m - i)$ channel vectors is $n - (m - i)$ dimensional. This means in each $\mathbf{h}'_i$, the “ISI-free” component $\mathbf{h}'_{i\perp}$ has $|\mathbf{h}'_{i\perp}|^2 \sim \chi^2_{2 \times (n-m+i)}$. In other words, after a similar coordination rotation, the vector $\mathbf{h}'_{i\perp}$ contains $(n - m + i)$ non-zero elements ([42]). In the

following sections, this observation leads to a conclusion regarding the diversity order of the output SNR in step i .

2. MRC After Nulling

Since MRC weighting is essential for the following analysis, a detailed derivation of the MRC weighting factors in [43] should be repeated here. This derivation also provides deep insight into how the V-BLAST coherently combines the independent data from the “orthogonalized branches.” Let’s express the MRC combining in (3.9) as:

$$\hat{s}'_i = \mathbf{w}_i^+ \mathbf{r}' = \mathbf{w}_i^+ \mathbf{h}'_i s'_i + \mathbf{w}_i^+ \boldsymbol{\xi}' = \boldsymbol{\alpha}_i^+ \mathbf{P}_i \mathbf{h}'_i s'_i + \boldsymbol{\alpha}_i^+ \mathbf{P}_i \boldsymbol{\xi}'$$

where $\boldsymbol{\alpha}_i^+$ is the combining coefficient. The projection matrix $\mathbf{P}_i$ has the property:

$$\begin{aligned} \mathbf{P}_i \mathbf{P}_i^+ &= \left[\mathbf{I} - \mathbf{H}_{(i)} (\mathbf{H}_{(i)} \mathbf{H}_{(i)}^+)^{-1} \mathbf{H}_{(i)}^+ \right] \cdot \left[\mathbf{I} - \mathbf{H}_{(i)} (\mathbf{H}_{(i)} \mathbf{H}_{(i)}^+)^{-1} \mathbf{H}_{(i)}^+ \right] \\ &= \left[\mathbf{I} - \mathbf{H}_{(i)} (\mathbf{H}_{(i)} \mathbf{H}_{(i)}^+)^{-1} \mathbf{H}_{(i)}^+ \right] - \mathbf{H}_{(i)} (\mathbf{H}_{(i)} \mathbf{H}_{(i)}^+)^{-1} \mathbf{H}_{(i)}^+ + \mathbf{H}_{(i)} (\mathbf{H}_{(i)} \mathbf{H}_{(i)}^+)^{-1} \mathbf{H}_{(i)}^+ \\ &= \mathbf{P}_i = \mathbf{P}_i^+ \end{aligned} \quad (3.15)$$

The output SNR after combining is

$$\begin{aligned} \gamma_i &= \frac{\boldsymbol{\alpha}_i^+ \mathbf{P}_i \mathbf{h}'_i s'_i \cdot s'^*_i \mathbf{h}_i'^+ \mathbf{P}_i^+ \boldsymbol{\alpha}_i}{\boldsymbol{\alpha}_i^+ \mathbf{P}_i \boldsymbol{\xi}' \boldsymbol{\xi}_i'^+ \mathbf{P}_i^+ \boldsymbol{\alpha}_i} = \frac{\boldsymbol{\alpha}_i^+ \mathbf{P}_i \mathbf{h}'_i \mathbf{h}_i'^+ \mathbf{P}_i^+ \boldsymbol{\alpha}_i |s'_i|^2}{\boldsymbol{\alpha}_i^+ \mathbf{P}_i \mathbf{P}_i^+ \boldsymbol{\alpha}_i \sigma_0^2} \\ &= \frac{(\mathbf{P}_i \boldsymbol{\alpha}_i)^+ (\mathbf{h}_i'^+ \mathbf{P}_i^+)^+ (\mathbf{h}_i'^+ \mathbf{P}_i^+) (\mathbf{P}_i \boldsymbol{\alpha}_i) |s'_i|^2}{(\mathbf{P}_i \boldsymbol{\alpha}_i)^+ (\mathbf{P}_i \boldsymbol{\alpha}_i) \sigma_0^2} = \frac{|(\mathbf{h}_i'^+ \mathbf{P}_i^+) (\mathbf{P}_i \boldsymbol{\alpha}_i)|^2 |s'_i|^2}{|(\mathbf{P}_i \boldsymbol{\alpha}_i)|^2 \sigma_0^2} \end{aligned} \quad (3.16)$$

where $|\boldsymbol{\xi}'_i|^2 = \sigma_0^2 \mathbf{I}$, assuming noise components are identical independent distributed (i.i.d.) at each stream. Maximizing the above SNR using Cauchy-Schwartz inequality leads to

$$\gamma_i = \frac{|(\mathbf{h}_i'^+ \mathbf{P}_i^+) (\mathbf{P}_i \boldsymbol{\alpha}_i)|^2 |s'_i|^2}{|(\mathbf{P}_i \boldsymbol{\alpha}_i)|^2 \sigma_0^2} \leq \frac{|(\mathbf{h}_i'^+ \mathbf{P}_i^+)|^2 \cdot |(\mathbf{P}_i \boldsymbol{\alpha}_i)|^2 |s'_i|^2}{|(\mathbf{P}_i \boldsymbol{\alpha}_i)|^2 \sigma_0^2} = \frac{|(\mathbf{P}_i \mathbf{h}'_i)|^2 |s'_i|^2}{\sigma_0^2} \quad (3.17)$$

Hence, the optimum combining factor $\alpha_i = \mathbf{h}'_i$, which is similar to the scheme in a traditional linear combination. Note the distribution property of the abovementioned random γ_i is only determined by chi-square distributed $|\mathbf{h}'_{i\perp}|^2$.

Furthermore, it can be proved that noise in different layer peeling steps is uncorrelated, because

$$\langle \mathbf{w}_i^* \xi'_i \xi''_j \mathbf{w}_j \rangle = \mathbf{h}_i^* \mathbf{P}_i \langle \xi''_i \xi''_j \rangle \mathbf{P}_j^* \mathbf{h}'_j = \mathbf{h}_i^* \mathbf{P}_i \mathbf{P}_j^* \mathbf{h}'_j \sigma_0^2 = |\mathbf{P}_i^* \mathbf{h}'_j|^2 \sigma_0^2 \delta_{ij} \quad (3.18)$$

where $\mathbf{P}_i$ - projection matrix is orthogonal to the subspace $[\mathbf{h}'_{i+1}, \dots, \mathbf{h}'_m]$ and $[\mathbf{h}'_{j+1}, \dots, \mathbf{h}'_m]$ and also has $\mathbf{P}_i \mathbf{P}_j = \mathbf{P}_i$, for $\forall j > i$.³ Similarly, the correlation of the different signal components is:

$$\langle \mathbf{w}_i^* \mathbf{h}'_i s'_i s'^*_j \mathbf{h}''_j \mathbf{w}_j \rangle = \mathbf{h}_i^* \mathbf{P}_i \langle \mathbf{h}'_i s'_i s'^*_j \mathbf{h}''_j \rangle \mathbf{P}_j^* \mathbf{h}'_j = |\mathbf{P}_i^* \mathbf{h}'_j|^2 |\mathbf{h}'_i s'_i|^2 \delta_{ij} \quad (3.19)$$

Consequently, the symbol error rate (SER) performance in each step is also independent.

3.3. Performance in ZF-VBLAST Steps

Obviously, that the performance evaluation in V-BLAST is much more complicated than any other conventional digital system. The complexity originates from:

- The stochastic nature of signals received over multi-path fading channels;
- Error propagations introduced by interference cancellation when there are error detections in previous steps;
- The block error rate (BLER) depending on each step's error rate; and
- The difference between the time average error rate and the instantaneous error rate

For these reasons, a rigorous performance terminology is indispensable to V-BLAST. Different performance measures are shown in Fig3.3. Their definitions in [43] are outlined here for

³ Because $\mathbf{P}_i$ is the projection matrix to the subspace orthogonal to the span $\{\mathbf{h}_{i+1}, \dots, \mathbf{h}_m\}$, which includes the subspace composed of $\{\mathbf{h}_{j+1}, \dots, \mathbf{h}_m\}$, $\mathbf{P}_i$ can also work as $\mathbf{P}_j$. Therefore, $\mathbf{P}_i \mathbf{P}_j = \mathbf{P}_i$ ([42]).

comparison purposes, although only two of them—conditional error probability $P_e(\gamma_i)$ and BLER P_B are studied in the following chapters.

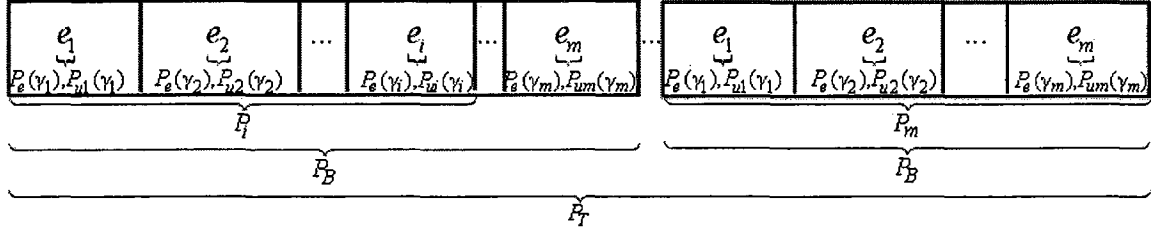


Fig. 3.3 Error rate symbols

Conditional Error Probability: $P_e(\gamma_i)$ is independent of the previous decoding steps, because “conditional” implies making an assumption that all the detections in the previous $i - 1$ steps are correct. $P_e(\gamma_i)$ is then only determined by instantaneous SNR at step i .

Unconditional Error Probability: $P_{ui}(\gamma_i)$ refers to the instantaneous error rate at step i no matter whether there are errors and how many errors occur in the previous steps. Any error detection during the previous steps will be propagated by $\mathbf{r}'_i = \mathbf{r} - \sum_{k=1}^{i-1} \mathbf{h}'_k \widehat{s'_k}$. Nulling with error propagation is given by:

$$\widehat{s'_i} = \mathbf{w}_i^+ \mathbf{r}' = \mathbf{a}_i^+ \left(\mathbf{P}_i \mathbf{h}'_i s'_i + \mathbf{P}_i \sum_{k=1}^{i-1} \mathbf{h}'_k \Delta s'_k + \mathbf{P}_i \boldsymbol{\xi} \right) \quad (3.20)$$

where $\Delta s'_k = s'_k - \widehat{s'_k}$. Thus, $P_{ui}(\gamma_i)$ not only relates to γ_i but also depends on the ordered detection queue. Note that a detection error does not necessarily cause the following errors, because it just boosts the noise level in (3.20). Usually, $P_{ui}(\gamma_i)$ can simply be bound by $P_e(\gamma_i) \leq P_{ui}(\gamma_i) \leq P_i$, where P_i is “the probability of having at least one error in steps 1 to i .”

Probability of Having At Least One Error During the Previous Steps: The probability of not having any error in steps 1 to i is $\prod_{k=1}^i (1 - P_e(\gamma_k))$. Thus, the probability of having at least one error can be found by

$$P_i = 1 - \prod_{k=1}^i (1 - P_e(\gamma_k)) = \sum_{k=1}^i P_e(\gamma_k) \prod_{l=1}^{k-1} (1 - P_e(\gamma_l)) \leq \sum_{k=1}^i P_e(\gamma_k) \quad (3.21)$$

Block Error Rate (BLER): P_B is the probability of having at least one symbol error in steps 1 to m , which equals

$$P_B = P_{i=m} = 1 - \prod_{k=1}^m (1 - P_e(\gamma_k)) = \sum_{k=1}^m P_e(\gamma_k) \prod_{l=1}^{k-1} (1 - P_e(\gamma_l)) \leq \sum_{k=1}^m P_e(\gamma_k) \quad (3.22)$$

Average Conditional Error Probability: The average SER of the decoding in the i th step is defined by

$$\overline{P_{e,i}} = \int_0^\infty P_e(\gamma_0 \cdot x) f_i(x) dx \quad (3.23)$$

where γ_0 is the average output SNR among steps and $x = \gamma_i / \gamma_0$. $P_e(\gamma_i)$ is the constellation-dependent instantaneous SER. Furthermore, because the distribution of the output SNR in step i is determined by $|\mathbf{h}'_{i\perp}|^2 \sim \chi^2_{2 \times (n-m+i)}$, the γ_i follows the probability density function (PDF) as (see Appendix A1 for the derivation): [43]

$$f_i(x) = x^{n-m+i-1} e^{-x} / (n-m+i-1)! \quad (3.24)$$

With (3.23) and (3.24), the average SER in each step i goes to

$$\overline{P_{e,i}}(\gamma_0) = \int_0^\infty P_e(\gamma_i) \frac{e^{-(\gamma_i/\gamma_0)} \gamma_i^{n-m+i-1}}{(n-m+i-1)! \gamma_0^{n-m+i}} d\gamma_i \quad (3.25)$$

Note (3.25) is exactly the same as that of the method in [50] used to derive the average SER for BPSK with MRC over multiple i.i.d. Rayleigh channels.

Average BLER: The average counterpart of (3.22) is

$$\overline{P}_B(\gamma_0) = \overline{P}_{i=m}(\gamma_0) = \sum_{k=1}^m \overline{P}_{e,k}(\gamma_0) \prod_{l=1}^{k-1} (1 - \overline{P}_{e,l}(\gamma_0)) \leq \sum_{k=1}^m \overline{P}_{e,k}(\gamma_0) \quad (3.26)$$

For the asymptotic cases—i.e., $\gamma_0 \gg 1$ —the instantaneous SER $P_e(\gamma_i/\gamma_0) = Q(\sqrt{k(\gamma_i/\gamma_0)\gamma_0})$ decreases exponentially when γ_0 increases. To this end, it can be expected that: 1) those small instantaneous γ_i with $\gamma_i/\gamma_0 \ll 1$ dominate the outage probabilities in each decoding step; and 2) BLER is dominated by the instantaneous SER in the first step, which suffers from the smallest diversity order. In this case, since $\overline{P}_e(\gamma_0) \ll 1$, there is an approximation of

$$\overline{P}_B(\gamma_0) = \overline{P}_{i=m}(\gamma_0) \approx \sum_{k=1}^m \overline{P}_{e,k}(\gamma_0) \approx \overline{P}_{e,1} \quad (3.27)$$

Name	Symbol	Definition
Conditional error rate at step i	$P_e(\gamma_i)$	The probability of having an error at step i given error free steps 1 to $(i - 1)$
Unconditional error rate at step i	$P_{ui}(\gamma_i)$	The probability of having an error at step i given the error propagation from the previous $i - 1$ steps
The probability of having at least one symbol error at step 1 to i	P_i	$P_i = 1 - \prod_{k=1}^i (1 - P_e(\gamma_k))$
Block error rate	P_B	The probability of having at least one symbol error in steps 1 to m
Average conditional error rate at step i	$\overline{P}_{e,i}$	Average probability of having an error at step i given error free steps 1 to $(i - 1)$ and average output SNR γ_0 .
Average BLER	$\overline{P}_B$	Average probability of having at least one symbol error in steps 1 to m given the average output SNR γ_0 among steps.
Total Error Probability	P_T	The average SER when the number of detected symbols goes to infinity

Table 3.1 Error rates and symbols ([43])

Total Error Probability: If the observation is extended over a long period of time—the number of detected symbols goes to infinity—the average SER over this period is $P_T = \lim_{N \rightarrow \infty} \frac{N_e}{N}$, where $N_e = \sum_{k=1}^m N_{e,k}$ and $N_{e,k}$ is the number of errors in step k . Considering the ergodic signal process at each step, the total error rate can be further expressed as

$$P_T = \frac{1}{m} \sum_{k=1}^m \overline{P_{u,k}} \quad (3.28)$$

where $\overline{P_{u,k}}$ is the average unconditional error probability at decoding step i over a long period of time. Table 3.1 provides a summary of error rate definitions for ZF-VBLAST in the thesis.

3.4. Summary

From the review of different nulling criteria in the first section of this chapter, the superiority of unordered ZF-VBLAST over the other VBLAST schemes should be clear. Of those alternatives to optimum ordering, non-uniform power allocation based on average BLER is attractive due to its simplicity. To find the optimum power allocation among the streams, an explicit target BLER expression is required. Thus, by summarizing the key ideas in [42] and [43] regarding γ_i distribution and V-BLAST performance evaluations, this chapter essentially set up a unifying framework (i.e., (3.25) and (3.27)) to evaluate the average BLER in an unordered ZF-VBLAST for various constellations. Replacing the specific BPSK instantaneous $P_e(\gamma_i)$ in (3.25) with a general one for various constellations is a promising strategy. This strategy helps us to attain a closed-form target BLER expression for the power allocation optimization.

Chapter 4. Average Union Bound (AUB) of SER

In this chapter, the key idea is to combine the well known expressions of SER for BPSK modulation with the nearest neighbor union bound of high order modulations to obtain a generic average SER expression for various modulation formats. The fact we demonstrated is that the union bound of average SER can be constructed by considering the dominating closest neighbors around each constellation point. Consequently, two parameters, N_c and D_s , which is closely related to the normalized minimum Euclidean distance d_{min}^* in the constellation, are used to quantify the above average bound.

4.1. Bound of Instantaneous SER

Given that a signal s_i is transmitted, the instantaneous SER of the system over an AWGN channel can be bounded by ([61]):

$$\mathcal{Q}\left(\sqrt{\frac{d_{min}^2}{2N_0}}\right) \leq P(e|s_i) \leq \sum_{j \neq i} \mathcal{Q}\left(\sqrt{\frac{d_{ij}^2}{2N_0}}\right) \quad (4.1)$$

where d_{ij} is the Euclidean distance between the constellation points s_i and s_j and d_{min} is the smallest Euclidean distance of any two constellation points in the constellation. An upper bound of the right-hand side of (4.1) may also be given as

$$P_e \leq (M-1) \cdot \mathcal{Q}\left(\sqrt{\frac{d_{min}^2}{2N_0}}\right) \quad (4.2)$$

Unfortunately, (4.2) is not tight. Considering only a few “closest neighbors,” which have a high influence on signal detection, the right hand side of (4.1) turns to

$$P_e \approx N_c \cdot \mathcal{Q}\left(\sqrt{\frac{d_{min}^2}{2N_0}}\right) \quad (4.3)$$

where N_c is the average number of “the nearest neighbors,” e.g., $N_c \approx 4$ in any high-order M-QAM given those boundary points have less than four nearest neighbors, yet $N_c = 2$ in 4-QAM. Notably, the approximation in (4.3) is due to the Gaussian tail probabilities corresponding to “overlapped” decision regions. To this end, we rewrite (4.3) as

$$P_e \approx N_c \cdot Q\left(\sqrt{D_s \cdot 2\gamma_s}\right) \quad (4.4)$$

where $D_s = d_{\min}^2/(4N_0\gamma_s) = d_{\min}^{*2}/4$, the normalized minimum Euclidean distance squared. (i.e., $d_{\min}^{*2}$ is the minimum Euclidean distance square “normalized” by energy per symbol $N_0\gamma_s$, so it is independent to γ_s (See AppendixA2). Both D_s and N_c are constellation-related constants.) Actually, [15] demonstrates that (4.3) is a tight bound. The validity of this bound is examined in the flowing examples of MPSK and MQAM.

3. BPSK

Obviously, the exact instantaneous BER (SER) of BPSK ([50])

$$P_e = Q\left(\sqrt{2\gamma_b}\right)$$

is the same as the bound defined by (4.4), with $N_c=1$ and $D_s=1$.

4. MPSK

The exact SER of MPSK is ([57])

$$P_e = \frac{1}{\pi} \int_0^{(M-1)\pi/M} \exp\left(-\gamma_s \frac{\sin^2(\pi/M)}{\sin^2\theta}\right) \cdot d\theta \quad (4.5)$$

Because each signal point has two nearest neighbors associated with $d_{\min}^* = 2 \sin(\pi/M)$ in MPSK constellations (see Appendix A2), the bound in (4.4) becomes

$$P_e \approx 2Q\left(\sqrt{2\gamma_s} \sin \frac{\pi}{M}\right) \quad (4.6)$$

which is exactly the same as the approximation given by [50] considering $N_c = 2$ and $D_s = \sin^2(\pi/M)$. Figure 4.1 shows the difference between the SER estimations by (4.5) and that (4.6) is neglectable in an 8-PSK case.

5. MQAM

The exact instantaneous SER of MQAM (when $M = 4^k$) is given by ([57])

$$P_e = \frac{4}{\pi} \left(\frac{\sqrt{M}-1}{\sqrt{M}} \right) \int_0^{\pi/2} \exp \left[-\gamma_s \frac{3}{2(M-1)\sin^2 \theta} \right] d\theta - \frac{4}{\pi} \left(\frac{\sqrt{M}-1}{\sqrt{M}} \right)^2 \int_0^{\pi/4} \exp \left[-\gamma_s \frac{3}{2(M-1)\sin^2 \theta} \right] d\theta \quad (4.7)$$

In the constellations of the optimum 8-QAM and 16-QAM, $d_{\min}^*{}^2 = 4/5$ in the 8-QAM constellation and $d_{\min}^*{}^2 = 2/5$ in 16-QAM (see Appendix A2). On the other hand, the average number of the closest neighbors is 9/4 in 8-QAM and 3 in 16-QAM separately. Hence, the AUB is qualified as

$$P_e \approx \frac{9}{4} Q \left(\sqrt{\frac{2\gamma_s}{5}} \right), 8\text{-QAM} \quad (4.8)$$

with $N_c = 9/4$, $D_s = 1/5$ and

$$P_e \approx 3Q \left(\sqrt{\frac{\gamma_s}{5}} \right), 16\text{-QAM} \quad (4.9)$$

given $N_c = 3$, $D_s = 1/10$.

The bound in (4.9) is exactly same as another P_e approximation for MQAM ($M = 4^k$) ([52]):

$$P_e \approx 4 \left(1 - \frac{1}{\sqrt{M}} \right) Q \left(\sqrt{\frac{3\gamma_s}{M-1}} \right) \quad (4.10)$$

In the cases where $M = 8$, as Figure 4.1 shows, both (4.8) and (4.10) are close approximations at moderate SNR. (Indeed, because of the exponentially decaying behavior of the Q-function and

the small difference in the argument of Q-function between (4.8) and (4.10), the asymptotic part of the instantaneous SER AUB deviates from the estimation given by (4.10).) Note that only deep fading corresponding to small instantaneous SNR affects the average SER. Thus, the union bound in (4.8) is applicable to our further analysis of the average SER over fading channels.

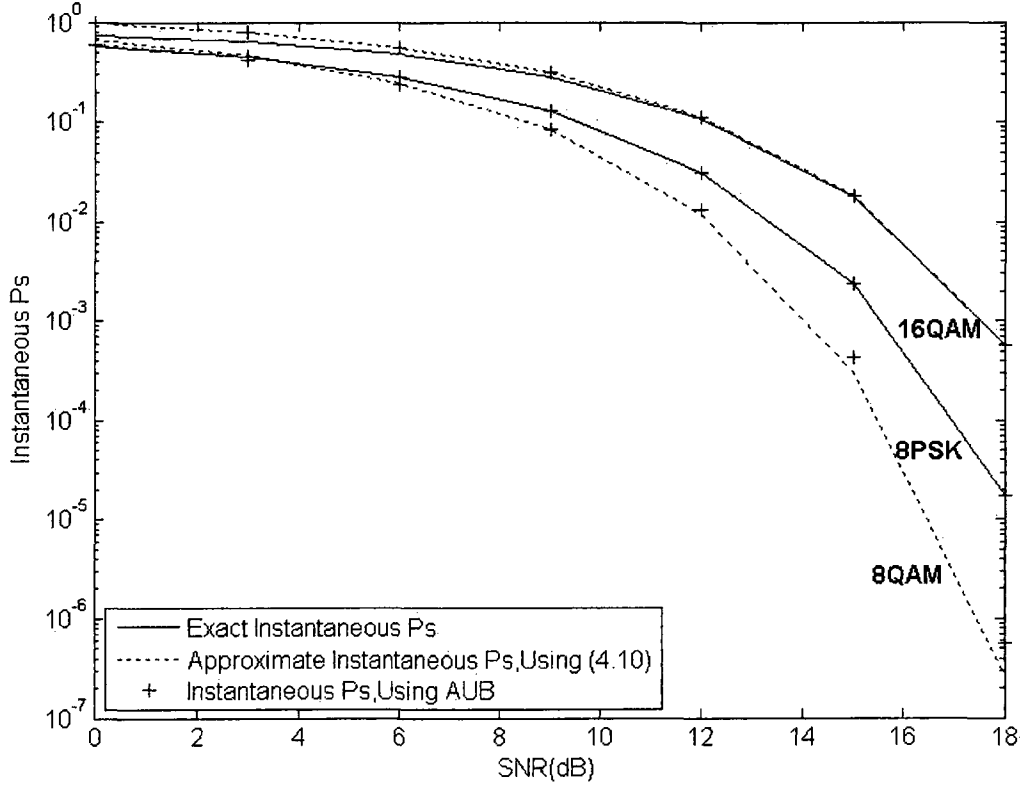


Fig. 4.1 Exact, approximate (4.10) and AUB of instantaneous SER vs. SNR

4.2. Bound of Average SER over Rayleigh Fading

The exact SER of BPSK over the Rayleigh fading channel is given by [50]

$$\bar{P}_e = \int_0^{\infty} Q(\sqrt{2\gamma_s}) \frac{1}{\gamma_0} e^{\frac{-\gamma_s}{\gamma_0}} d\gamma_s = \frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0}{1+\gamma_0}} \right)$$

where γ_0 is the average SNR per symbol.

The above equation can be generalized to higher level modulations by replacing the BPSK instantaneous SER $P_e = Q(\sqrt{2\gamma_s})$ with the AUB $P_e \approx N_c \cdot Q(\sqrt{D_s \cdot 2\gamma_s})$

$$\overline{P_e} \approx \int_0^\infty N_c Q(\sqrt{D_s \cdot 2\gamma_s}) \frac{1}{\gamma_0} e^{-\frac{\gamma_s}{\gamma_0}} d\gamma_s \xrightarrow{D_s \cdot \gamma_s = \gamma_s'} N_c \int_0^\infty Q(\sqrt{2\gamma_s'}) \frac{1}{D_s \cdot \gamma_0} e^{-\frac{\gamma_s'}{D_s \cdot \gamma_0}} d\gamma_s'$$

Therefore,

$$\overline{P_e} \approx N_c \cdot \frac{1}{2} \left(1 - \sqrt{\frac{D_s \cdot \gamma_0}{1 + D_s \cdot \gamma_0}} \right) \quad (4.11)$$

The specific expressions for various modulations are investigated below.

1. MPSK

The exact SER over Rayleigh fading is given by ([57])

$$\overline{P_e} = \left(\frac{M-1}{M} \right) \left\{ 1 - \sqrt{\frac{\sin^2(\pi/M)\gamma_0}{1 + \sin^2(\pi/M)\gamma_0}} \frac{M}{(M-1)\pi} \times \left[\frac{\pi}{2} + \tan^{-1} \left(\sqrt{\frac{\sin^2(\pi/M)\gamma_0}{1 + \sin^2(\pi/M)\gamma_0}} \cot \frac{\pi}{M} \right) \right] \right\}$$

It can be upper bounded by (4.11) with the coefficients $N_c = 2$ and $D_s = \sin^2(\pi/M)$ as in (4.6)

$$\overline{P_e} \approx 1 - \sqrt{\frac{\sin^2(\pi/M) \cdot \gamma_0}{1 + \sin^2(\pi/M) \cdot \gamma_0}} \quad (4.12)$$

Figure 4.2a shows that (4.12) gives an upper-bound tightly.

2. MQAM

The exact SER expression is given by (when $M = 4^k$) ([57])

$$\overline{P_e} = 2 \left(\frac{\sqrt{M}-1}{\sqrt{M}} \right) \left(1 - \sqrt{\frac{1.5\gamma_0}{M-1+1.5\gamma_0}} \right) - \left(\frac{\sqrt{M}-1}{\sqrt{M}} \right)^2 \left\{ 1 - \sqrt{\frac{1.5\gamma_0}{M-1+1.5\gamma_0}} \frac{4}{\pi} \tan^{-1} \left(\sqrt{\frac{M-1+1.5\gamma_0}{1.5\gamma_0}} \right) \right\}$$

It is apparent that the AUB for the optimum 8-QAM and 16-QAM are:

$$\overline{P_{e,8\text{-QAM}}} \approx \frac{9}{4} \cdot \frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0/5}{1 + \gamma_0/5}} \right) \quad (4.13)$$

$$\overline{P}_{e,16\text{-QAM}} \approx 3 \cdot \frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0/10}{1 + \gamma_0/10}} \right) \quad (4.14)$$

If the approximation of the instantaneous SER in (4.10) is used, (i.e., $N_c = 4(1 - 1/\sqrt{M})$ and $D_s = 3\gamma_0/[2(M-1)]$), (4.14) can be generalized to

$$\overline{P}_e \approx 4 \left(1 - \frac{1}{\sqrt{M}} \right) \cdot \frac{1}{2} \left(1 - \sqrt{\frac{3\gamma_0/[2(M-1)]}{1 + 3\gamma_0/[2(M-1)]}} \right) \quad (4.15)$$

for $M = 4^k$ cases.

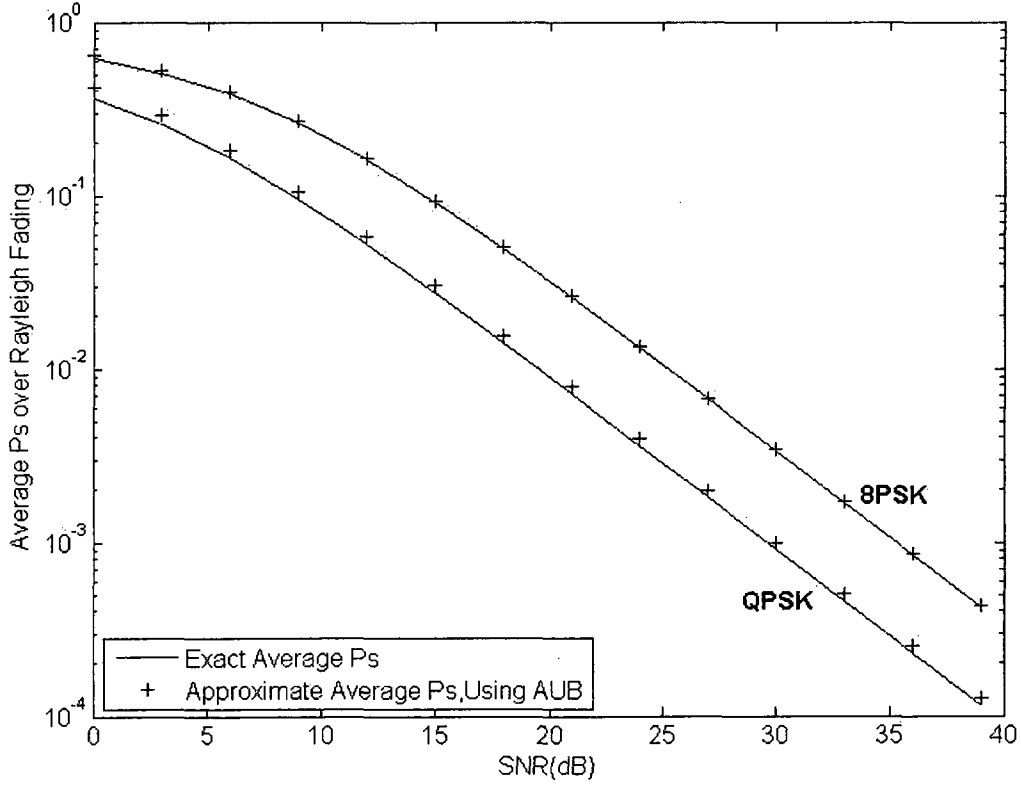


Fig. 4.2a Exact and AUB of average SER (over Rayleigh fading, MPSK) vs. SNR

As the exact SER expression of 8-QAM is unknown, the Monte Carlo (MC) simulation results of the 8-QAM over Rayleigh fading are outlined in Figure 4.2b as a baseline. In the 8-

QAM case, Figure 4.2b demonstrates that both bounds in (4.13) and (4.15) can closely approximate the MQAM performance even when $M \neq 4^k$.

Because the errors caused by deep fading (i.e., the small instantaneous SNR) dominate the average SER and AUB is accurate when instantaneous SNR is small (Figure 4.1), the difference between the AUB and the simulation baseline is only $\pm 20\%$, as Figure 4.2b shows. The gap does not become wider as the average SNR increases. In fact, (4.15) shows us a generic AUB expression for MQAM ($M = 4^k$) cases.

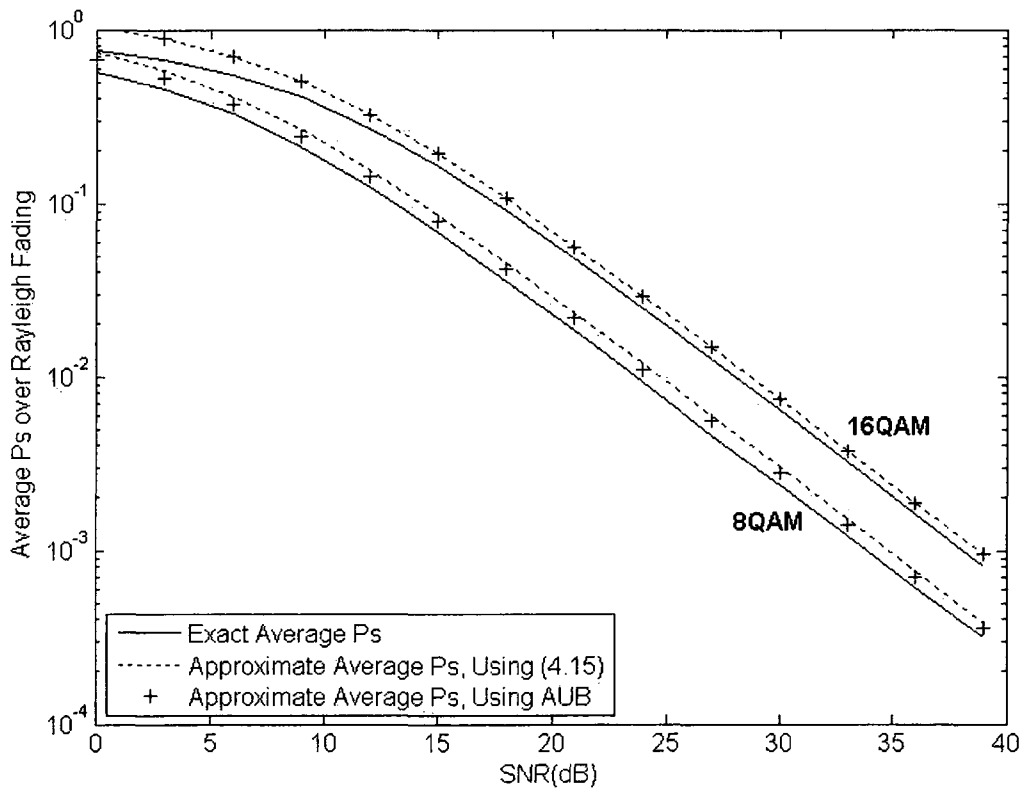


Fig. 4.2b Exact, approximation (4.15) and AUB average SER (over Rayleigh fading, MQAM) vs. SNR

4.3. Bound of SER over Diversity Rayleigh Channels

As the above examples depict the AUB as a valid estimation of system performance over a Rayleigh fading channel, it is conceivable that this bound is also credible in a diversity fading scenario. The exact average SER of BPSK with MRC over L i.i.d. Rayleigh channels is ([50])

$$\overline{P_e} = \int_0^\infty P_e(\gamma_b) p(\gamma_b) d\gamma_b = \int_0^\infty Q(\sqrt{2\gamma_b}) \frac{1}{(L-1)! \gamma_0^L} (\gamma_b)^{L-1} e^{-\frac{\gamma_b}{\gamma_0}} d\gamma_b = \left[\frac{1}{2}(1-\mu) \right]^L \sum_{k=0}^{L-1} \binom{L-1+k}{k} \left[\frac{1}{2}(1+\mu) \right]^k$$

where

$$\mu = \sqrt{\frac{\gamma_0}{1+\gamma_0}}$$

By substituting the bound of various modulations $P_e \approx N_c \cdot Q(\sqrt{D_s \cdot 2\gamma_s})$ into the above closed-form expression, we obtain

$$\begin{aligned} \overline{P_e} &\approx N_c \int_0^\infty Q(\sqrt{D_s \cdot 2\gamma_s}) \frac{1}{(L-1)! \gamma_0^L} (\gamma_s)^{L-1} e^{-\frac{\gamma_s}{\gamma_0}} d\gamma_s \\ &\xrightarrow{D_s \gamma_s = \gamma_s'} N_c \int_0^\infty Q(\sqrt{2\gamma_s'}) \frac{1}{(L-1)! D_s \cdot \gamma_0^L} \frac{(\gamma_s')^{L-1}}{D_s^{L-1}} e^{-\frac{\gamma_s'}{D_s \gamma_0}} d\gamma_s' \\ &= N_c \left[\frac{1}{2}(1-\mu') \right]^L \sum_{k=0}^{L-1} \binom{L-1+k}{k} \left[\frac{1}{2}(1+\mu') \right]^k \end{aligned} \quad (4.16)$$

where

$$\mu' = \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}}$$

In this section, in addition to comparing the AUB to the exact average SER, another well-validated asymptotic approximation of SER from [68] is used in for comparison purposes. This asymptotic model is explained below.

For the SER behavior of a system over diversity Rayleigh channel with MRC, there is an asymptotic approximation, which is

$$\overline{P_e} \approx (G\gamma_0)^{-D}$$

where G and D are denoted as “coding gain” and “diversity gain” respectively and given by

$$\begin{cases} G = 2D_s \left(N_c \frac{2^{L-1} \Gamma(L+0.5)}{\sqrt{\pi} L!} \right)^{-\frac{1}{L}} \\ D = L \end{cases}$$

Thus, the asymptotic approximation of the average SER turns into

$$\overline{P_e} \approx (G\gamma_0)^{-D} = \frac{N_c}{2\sqrt{\pi}} \left(\frac{1}{D_s \gamma_0} \right)^L \frac{\Gamma(L+0.5)}{\Gamma(L+1)} \quad (4.17)$$

where the diversity gain increases the slope of the SER curve, while the coding gain shifts the curve to the left. A detail introduction to this asymptotic model (Wang’s model) is given in Appendix A3. Note that the original model is not parameterized with the constellation-dependent parameters D_s and N_c in the paper. Appendix A3 shows how to generalize it to (4.17) for arbitrary modulation format with D_s and N_c .

1. MPSK

The AUB of MPSK can be obtained from (4.6) and (4.16):

$$\overline{P_e} \approx N_c \left[\frac{1}{2} \left(1 - \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}} \right) \right]^L \sum_{k=0}^{L-1} \binom{L-1+k}{k} \left[\frac{1}{2} \left(1 + \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}} \right) \right]^k \quad (4.18)$$

where

$$\begin{cases} N_{c,\text{MPSK}} = 2 \\ D_{s,\text{MPSK}} = \sin^2 \frac{\pi}{M} \end{cases}$$

Figure 4.3a compares the SER performance estimated by (4.17) and (4.18), denoted by “Asymptotic Approximation” and “AUB.” In Figure 4.3a, the numerical SER performance is also outlined with ([57])

$$\overline{P_e} = \frac{1}{\pi} \int_0^{(M-1)\pi/M} \left(1 + \frac{\sin^2(\pi/M)}{\sin^2 \varphi} \gamma_0 \right)^{-L} d\varphi \quad (4.19)$$

Comparing to the asymptotic approximation (4.17), we may find that the advantage of AUB we are using is that it provides good accuracy at low and intermediate SNR as well.

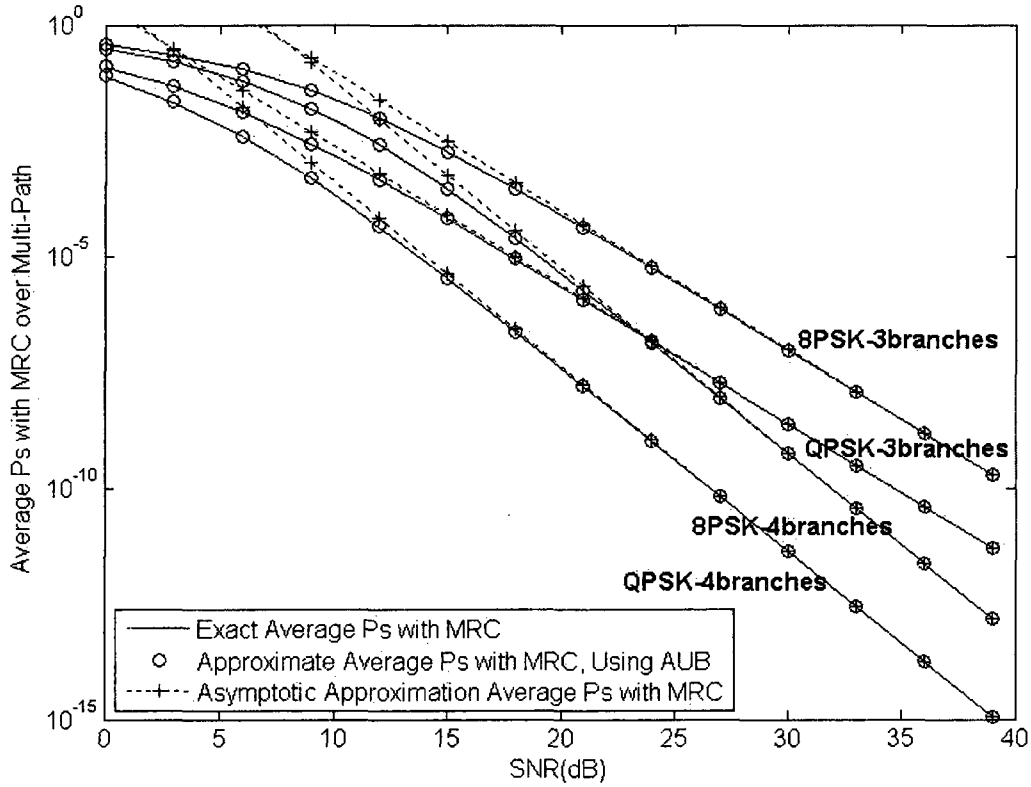


Fig. 4.3a Exact, asymptotic approximation (4.17) and AUB average SER (MPSK over 3 and 4 i.i.d. Rayleigh branches with MRC) vs. SNR

2. MQAM

Similarly, the upper bounds in MQAM cases are approximated by

$$\overline{P_e} \approx N_c \left[\frac{1}{2} \left(1 - \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}} \right) \right]^L \sum_{k=0}^{L-1} \binom{L-1+k}{k} \left[\frac{1}{2} \left(1 + \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}} \right) \right]^k$$

where

$$\begin{cases} N_{c,8QAM} = 9/4 \\ D_{s,8QAM} = 1/5 \end{cases} \quad (4.20)$$

and

$$\begin{cases} N_{c,16QAM} = 3 \\ D_{s,16QAM} = 1/10 \end{cases} \quad (4.21)$$

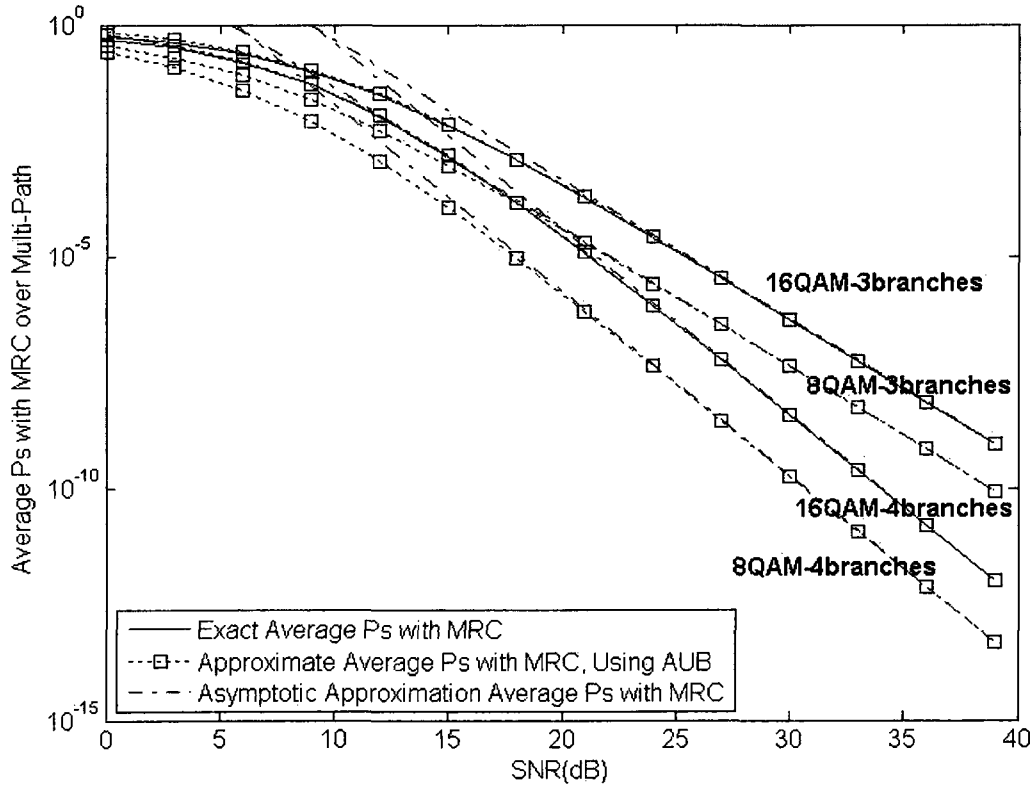


Fig. 4.3b Exact, asymptotic approximation (4.17) and AUB average SER (MQAM over 3 and 4 i.i.d. Rayleigh branches with MRC) vs. SNR

Again, if the general $P_e \approx 4 \left(1 - \frac{1}{\sqrt{M}}\right) Q \left(\sqrt{\frac{3\gamma_s}{M-1}} \right)$ in (4.10) is substituted into (4.16), the constants

D_s and N_c change to

$$\begin{cases} N_{c, \text{MQAM}} = 4(1 - 1/\sqrt{M}) \\ D_{s, \text{MQAM}} = 3/[2(M-1)] \end{cases}, M = 4^k \quad (4.22)$$

Figure 4.3b shows the average SER estimated by AUB in (4.20)~(4.21) and Wang's asymptotic approximation in (4.17) with $(D_{s, \text{MQAM}}, N_{c, \text{MQAM}})$ in (4.22)—denoted as “AUB” and “Asymptotic Approximation” separately. The exact SER in Figure 4.3b is given by ([57])

$$\overline{P_e} = \frac{4}{\pi} \left(1 - \frac{1}{\sqrt{M}}\right) \int_0^{\pi/2} \left(1 + \frac{3\gamma_0}{2(M-1)\sin^2 \varphi}\right)^{-L} d\varphi - \frac{4}{\pi} \left(1 - \frac{1}{\sqrt{M}}\right)^2 \int_0^{\pi/4} \left(1 + \frac{3\gamma_0}{2(M-1)\sin^2 \varphi}\right)^{-L} d\varphi \quad (4.23)$$

for $M = 4^k$.

It is apparent that the AUB indicated in Figure 4.3 are tight. For example, in 8-QAM, the worst case in our analysis, the performance gap between the union bound (4.20) and the rigorously conducted approximation (4.17) is extremely narrow; i.e., the difference is $\pm 8\%$ and $\pm 15\%$ when $\gamma_0 > 30$ dB for 3-branches and 4-branches diversity combining separately. The inaccuracy increases slightly with L and γ_0 for the following reason: The diversity combining output SNR γ_s is chi-square distributed. The probability of having large γ_s increases when the mean $L\gamma_0$ increases. As we discussed in section 4.1, the higher instantaneous SNR γ_s we have, the less accurate this AUB estimation is. The inaccuracy is due to the exponentially decaying behavior of the Q-function and the small difference in the argument of Q-function between (4.8) and (4.10).

The paradigms in Figure 4.3 also show that although both the AUB and Wang's parameterization achieve satisfying asymptotic approximations, the former is superior to the latter at the small or moderate SNR.

4.4. Discussion

The above analysis shows that the average SER expression of the BPSK system over diversity Rayleigh fading channels offers a closed-form template for the estimation of average SER for higher-level modulations. The AUB, on the other hand, explains why it is feasible: the instantaneous SER of different modulations can be represented in similar Q-function form by considering nearest constellation neighbors only. In these forms, instantaneous SER is parameterized with two constellation-dependent coefficients; D_s , the normalized minimum Euclidean distance d_{min}^* ; and N_c , the average number of the closest neighbors. The general expressions of this AUB are given by

$$\overline{P}_{e, \text{Rayleigh}} \approx N_c \cdot \frac{1}{2} \left(1 - \sqrt{\frac{D_s \cdot \gamma_0}{1 + D_s \cdot \gamma_0}} \right)$$

$$\overline{P}_{e, \text{MRC}} \approx N_c \left[\frac{1}{2} \left(1 - \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}} \right) \right]^L \sum_{k=0}^{L-1} \binom{L-1+k}{k} \left[\frac{1}{2} \left(1 + \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}} \right) \right]^k$$

where the coefficients can be summarized in Table. 4.1.

	D_s	N_c
BPSK	1	1
MPSK ($M \geq 4$)	$\sin^2(\pi / M)$	2
8QAM	1 / 5	9 / 4
16QAM	1 / 10	3
MQAM	$3 / [2 \times (M - 1)]$	$4 \times (1 - 1 / M^{1/2})$
Others	$d_{min}^2 / (2N_0 \times 2\gamma_s)$	average number of the closest neighbors

Table. 4.1 Constellation-related parameters D_s and N_c for MPSK and MQAM

4.5. Summary

Of the different fading scenarios, the AUB of SER proves to be a universal approximation for various modulations. To illustrate the accuracy of this bound, MC simulations, exact performance formulas and verified asymptotic approximation were employed. Most significantly, the abovementioned insightful union bound approach not only reveals which factors affect diversity system performance, but also shows how these factors affect it. From the point of view of this thesis, the appealing point is that this bound makes the cumbersome expressions of the diversity systems SER performance succinct and significantly simplifies the derivations of our optimum power allocations in V-BLAST cases below.

Chapter 5. Optimum Power Allocation

Of the variety of approaches aimed at minimizing the error rate in the original V-BLAST algorithm, as discussed in Chapter 2, non-uniform power allocation among the transmit streams is computationally effective and instrumental in attenuating error propagation from the lower detection steps. In [32], comparing the two optimum power allocation criteria—instantaneous optimum allocation based on TBER and average optimum allocation based on BLER—results in a preference for the latter as being more computationally effective from a practical point of view. However, the analysis of optimum power allocation done by [32] was limited for BPSK. Thus, this average optimum power allocation is generalized here for higher order modulations. For the generalization, because the i th detection step in an $m \times n$ unordered V-BLAST exhibits $(n - m + i)$ -th order diversity (Chapter 3), the average union SER bound of diversity systems in Chapter 4 can be adopted to generalize the average BLER-based non-uniform power allocation proposed in [32]

5.1. Block Error Rate of V-BLAST

1. Union Bound of $\bar{P}_{ei}$

In an MRC-weighted V-BLAST, the distribution properties of the SNR at the i th step are given by [43] and explained in Chapter 3. As this instantaneous SNR γ_i follows the probability density function (PDF) $f_i(\gamma_i / \gamma_0) = x^{n-m+i-1} e^{-\gamma_i / \gamma_0} / (n - m + i - 1)!$, the conditional $\bar{P}_{e,i}(\gamma_i)$ (i.e., conditioned on previous error-free detections) is same as the SER of an $(n - m + i)$ -th order MRC system. Replacing the diversity order L in (4.16) with $n - m + i$, a general closed-form approximation of $\bar{P}_{e,i}$ is obtained:

$$\begin{cases} \overline{P}_{ei}(\gamma_i) \approx N_c \left[\frac{1}{2}(1 - \mu_i') \right]^{n-m+i} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \left[\frac{1}{2}(1 + \mu_i') \right]^k \\ \mu_i' = \sqrt{\frac{D_s \gamma_0}{1 + D_s \gamma_0}} \end{cases} \quad (5.1)$$

where

$$\begin{cases} N_{c,M\text{-PSK}} = 2; D_{s,M\text{-PSK}} = \sin^2 \frac{\pi}{M} \\ N_{c,M\text{-QAM}} = 4 \left(1 - \frac{1}{\sqrt{M}} \right); D_{s,M\text{-QAM}} = \frac{3}{2(M-1)} \end{cases} \quad (5.2)$$

for general MPSK and MQAM ($M = 4^k$).

2. Union bound of $\overline{P}_B$

In Chapter 3, the asymptotic total BLER can be approximated by the summation of $\overline{P}_{e,i}$:

$$\overline{P}_B = 1 - \prod_{i=1}^m (1 - \overline{P}_{ei}(\gamma_i)) \approx \sum_i \overline{P}_{ei}(\gamma_i)$$

With the union bound of $\overline{P}_{e,i}$ in (5.2), this becomes

$$\overline{P}_B \approx \sum_{i=1}^m \overline{P}_{ei} \approx \sum_{i=1}^m N_c \left[\frac{1}{2}(1 - \mu_i') \right]^{n-m+i} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \left[\frac{1}{2}(1 + \mu_i') \right]^k \quad (5.3)$$

assuming each transmitter branch has the identical average SNR γ_0 .

Note that the rest of this thesis is based on the above average BLER approximation instead of the average TBER, because 1) the derivation of TBER requires explicit quantifications of the error propagations between error patterns $\mathbf{E}_i$ among the steps and the complexity increases over the steps; 2) [32] proves the optimization based on average BLER can result in a unique “global” solution for different modulations, but the uniqueness of the solution based on average TBER really depends on the statistics of $\mathbf{H}$.

5.2. Optimum Power Allocation for V-BLAST

1. Optimum Power Allocation for high SNR

In Chapter 3, we demonstrated that the first step in the original V-BLAST decoding shows the smallest diversity order and then suffers the biggest $\bar{P}_{ei}$ in the high SNR region, which dominates the $\bar{P}_B$. Hence, more power on the first and second transmitter branches (i.e., the lower steps) is preferable for the average BLER. For this reason, if the non-uniform power allocation coefficient on the i th branch is denoted as α_i , (5.3) becomes (see Appendix A4)

$$\bar{P}_B(\mathbf{\alpha}) \approx N_c \sum_{i=1}^m \frac{C_{2(n-m+i)-1}^{n-m+i}}{(4\alpha_i \cdot D_s \gamma_0)^{n-m+i}} \quad (5.4)$$

Under the total power constraint

$$\sum_{i=1}^m \alpha_i = m$$

the minimum of the objective function $\bar{P}_B(\mathbf{\alpha})$ can be derived via the following Lagrange equation:

$$L(\mathbf{\alpha}) = \bar{P}_B(\mathbf{\alpha}, \gamma_0) + \lambda \left(\sum_{i=1}^m \alpha_i - m \right) \quad (5.5)$$

The derivative of this Lagrange equation regarding the allocation coefficient α_i is

$$\frac{\partial L(\mathbf{\alpha})}{\partial \alpha_i} = -N_c \cdot \frac{n-m+i}{(\alpha_i)^{n-m+i+1}} \frac{C_{2(n-m+i)-1}^{n-m+i}}{(4D_s \gamma_0)^{n-m+i}} + \lambda$$

The optimum $\mathbf{\alpha}_{\text{opt}}$ can be found when this derivative equals zero. Appendix.A5 shows that $\mathbf{\alpha}_{\text{opt}}$ is given by

$$\tilde{\alpha}_{i,\text{opt}} \approx b_i (4D_s \gamma_0)^{(1-i)/(n-m+i+1)} \left[1 - \frac{b_2}{m c_1^{n-m+3} \sqrt[3]{4D_s \gamma_0}} \right]^{c_i}$$

with the normalized ones:

$$\begin{cases} \alpha_{i,\text{opt}} = m \tilde{\alpha}_{i,\text{opt}} / \sum_{k=1}^m \tilde{\alpha}_{k,\text{opt}} & i > 1 \\ \alpha_{1,\text{opt}} = m - \sum_{i=2}^m \alpha_{i,\text{opt}} \end{cases} \quad (5.6)$$

where

$$\begin{aligned} a_i &= {}^{n-m+i+1}\sqrt{N_c (n-m+i) C_{2(n-m+i)-1}^{n-m+i}} \\ b_i &= {}^{n-m+i+1}\sqrt{\frac{(n-m+i) C_{2(n-m+i)-1}^{n-m+i} \cdot m^{n-m+2}}{(n-m+1) C_{2(n-m+1)-1}^{n-m+1}}} \\ c_i &= \frac{(n+1)!}{(n-m+1)!(n-m+i+1)} \end{aligned}$$

and the Lagrange multiplier is

$$\lambda \approx \frac{1}{(4D_s \gamma_0)^{(n-m+1)}} \left(\frac{a_1}{m} \right)^{n-m+2} \left[1 - \frac{b_2}{m c_1^{n-m+3} \sqrt{4D_s \gamma_0}} \right]^{-(n-m+2)c_1} \quad (5.7)$$

Given the condition of $b_2 / (m c_1^{n-m+3} \sqrt{4D_s \gamma_0}) \ll 1$ for high SNR, the further approximation of (5.6) is obtained:

$$\begin{cases} \tilde{\alpha}_{i,\text{opt}} \approx b_i (4D_s \gamma_0)^{(1-i)/(n-m+i+1)} \\ \alpha_{i,\text{opt}} = m \tilde{\alpha}_{i,\text{opt}} / \sum_{k=1}^m \tilde{\alpha}_{k,\text{opt}} \\ \alpha_{1,\text{opt}} = m - \sum_{i=2}^m \alpha_{i,\text{opt}} \end{cases} \quad (5.8)$$

and the average BLER in (5.4) with α_{opt} then becomes

$$\overline{P_B} \approx N_c \sum_{i=1}^m \frac{C_{2(n-m+i)-1}^{n-m+i}}{b_i^{n-m+i}} \left(\frac{1}{4D_s \gamma_0} \right)^{(n-m+i) \left(\frac{n-m+2}{n-m+i+1} \right)} \quad (5.9)$$

Let us compare this approximation to the performance in (5.4) when $\alpha_i = 1$. It is clear that the diversity order of the higher steps ($i > 1$) in the power optimized V-BLAST is less than the

counterpart in an un-optimized V-BLAST, where $\bar{P}_{e,i} \propto (1/4D_s\gamma_0)^{n-m+i}$. For example, in the first step, the diversity orders $d_{1,\text{un-optimized}}$ for un-optimized and $d_{1,\text{optimized}}$ for optimized V-BLAST are

$$\begin{cases} d_{1,\text{un-optimized}} = n - m + 1 \\ d_{1,\text{optimized}} = n - m + 1 \end{cases}$$

and in the higher steps ($i > 1$), the diversity orders are

$$\begin{cases} d_{i,\text{un-optimized}} = n - m + i \\ d_{i,\text{optimized}} = (n - m + i) \left(\frac{n - m + 2}{n - m + i + 1} \right) \end{cases}$$

Therefore, $d_{i,\text{un-optimized}} > d_{i,\text{optimized}} > d_{1,\text{optimized}}$ and the first step still dominates the total BLER.

In (5.9), since the coefficient $b_1 = m$, the new $\bar{P}_{e,1}$ is $1/m$ of the $\bar{P}_{e,1}$ of the un-optimized V-BLAST when $m = n$. Considering the fact that $\bar{P}_{e,1} \gg \bar{P}_{e,2}$ and $\bar{P}_{e,1}$ dominates $\bar{P}_B$, it can be said that the asymptotic $\bar{P}_B$ is reduced by roughly m times. Additionally, there are two observations about the influence of the modulation-dependent factor $1/D_s$ on BLER:

- The non-uniform power optimization improves $\bar{P}_{e,1}$ but leaves the remaining $\bar{P}_{e,i}$ ($i > 1$) adverse. To this end, in the high steps of a higher-level constellation, where D_s is smaller than the D_s 's in lower-level modulations, the $\bar{P}_{e,i}$ ($i > 1$) in optimized higher-level modulations will be higher than the counterpart in optimized lower-level modulations.

- The fractional diversity order $d_{i,\text{optimized}} = (n - m + i) \left(\frac{n - m + 2}{n - m + i + 1} \right)$ in optimized higher steps is smaller than $d_{i,\text{un-optimized}} = n - m + i$ in an un-optimized system. Therefore, comparing the $\bar{P}_{e,i}$ of optimized and un-optimized systems,

$$\begin{cases} \overline{P}_{e,i \text{ un-optimized}} \approx N_c \frac{C_{2(n-m+i)-1}^{n-m+i}}{(4D_s\gamma_0)^{n-m+i}} \\ \overline{P}_{e,i \text{ optimized}} \approx N_c \frac{C_{2(n-m+i)-1}^{n-m+i}}{b_i^{n-m+i}} \left(\frac{1}{4D_s\gamma_0} \right)^{(n-m+i)\left(\frac{n-m+2}{n-m+i+1}\right)} \end{cases} \quad (5.10)$$

one may have $\overline{P}_{e,i \text{ un-optimized}} / \overline{P}_{e,i \text{ optimized}} = b_i^{n-m+i} / (4D_s\gamma_0)^{(n-m+i)\left(\frac{n-m+2}{n-m+i+1}\right)} < 1$.

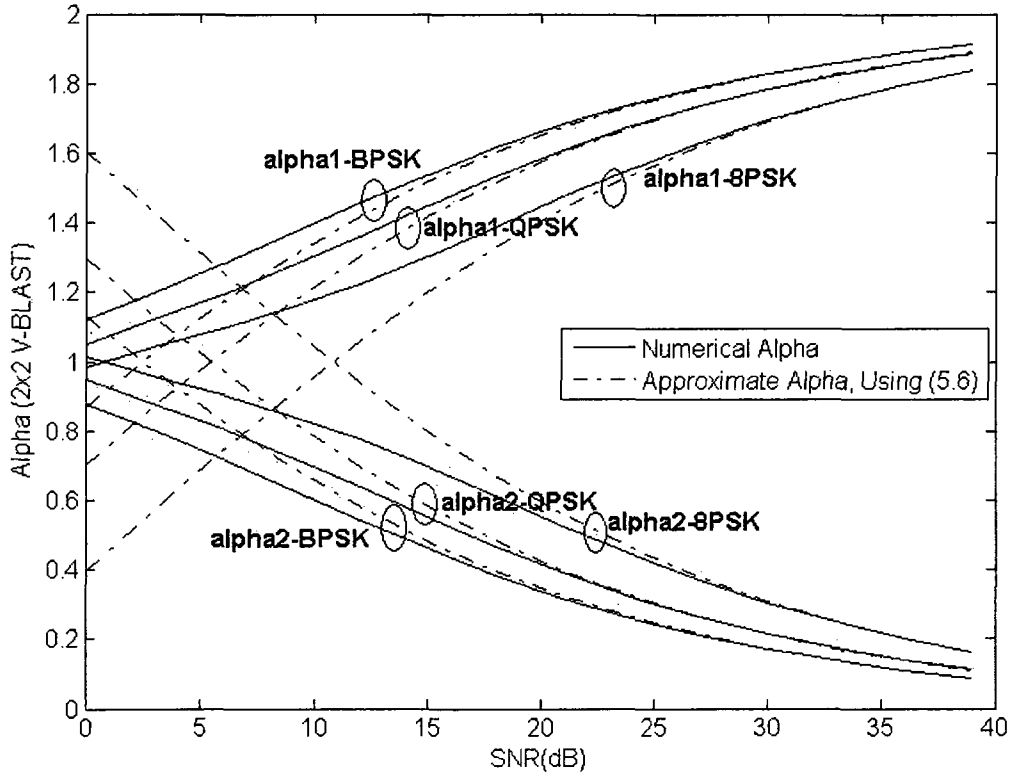


Fig. 5.1a Numerical α_{opt} and approximate α_{opt} (5.6) (for MPSK, 2 by 2 ZF V-BLAST) vs. SNR

After all, in asymptotic cases, when $\overline{P}_{e,1}$ dominates $\overline{P}_B$, the optimization of (5.6) can still achieve performance improvements and (5.9) can be further approximated as

$$\overline{P}_B(\alpha_{\text{opt}}) \approx N_c \frac{1}{(4mD_s\gamma_0)^{n-m+1}} \quad (5.11)$$

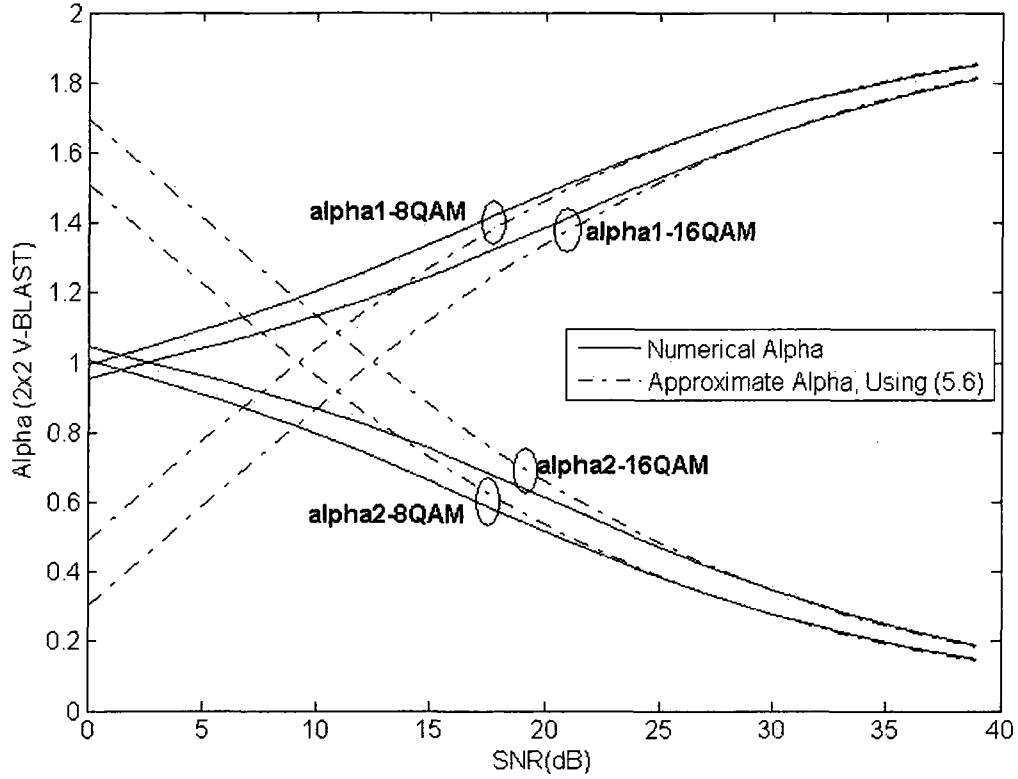


Fig. 5.1b Numerical α_{opt} and approximate α_{opt} (5.6) (for MQAM, 2 by 2 ZF V-BLAST) vs. SNR

Figure 5.1 compares the approximate α_{opt} in (5.6) with the exact α_{opt} found numerically with the Sequential Quadratic Programming (SQP) function, “fmincon,” in Matlab®. (The former is referred to as “approximate” and the latter is “numerical” in the diagram.) The conclusion is that (5.6) only works for the high SNR cases. In the moderate SNR region, the higher the level of the modulation is, the less accurate the approximation in (5.6) is. Certainly, as (5.8) shows, the higher constellation level, which is parameterized by a smaller D_s , amounts to a slower convergence of α_i ($i > 1$) to zero; i.e., a slower convergence of α_1 to m .

In fact, comparing the BLER performance estimated by MC simulations in Figure 5.2, the quasi-optimization with the α_{opt} in (5.6) “Approximate Optimization” still performs equally well

as the optimization with the numerical α_{opt} “Numerical Optimization”. (Note that the “analytical”

BLER in Figure 5.2 is $\bar{P}_B = 1 - \prod_{i=1}^m (1 - \bar{P}_{ei}(\gamma_i))$ where $\bar{P}_{ei}$ comes from the exact SER (4.19) and

(4.23) while $\bar{P}_{ei,8QAM}$ is AUB from (4.20).)

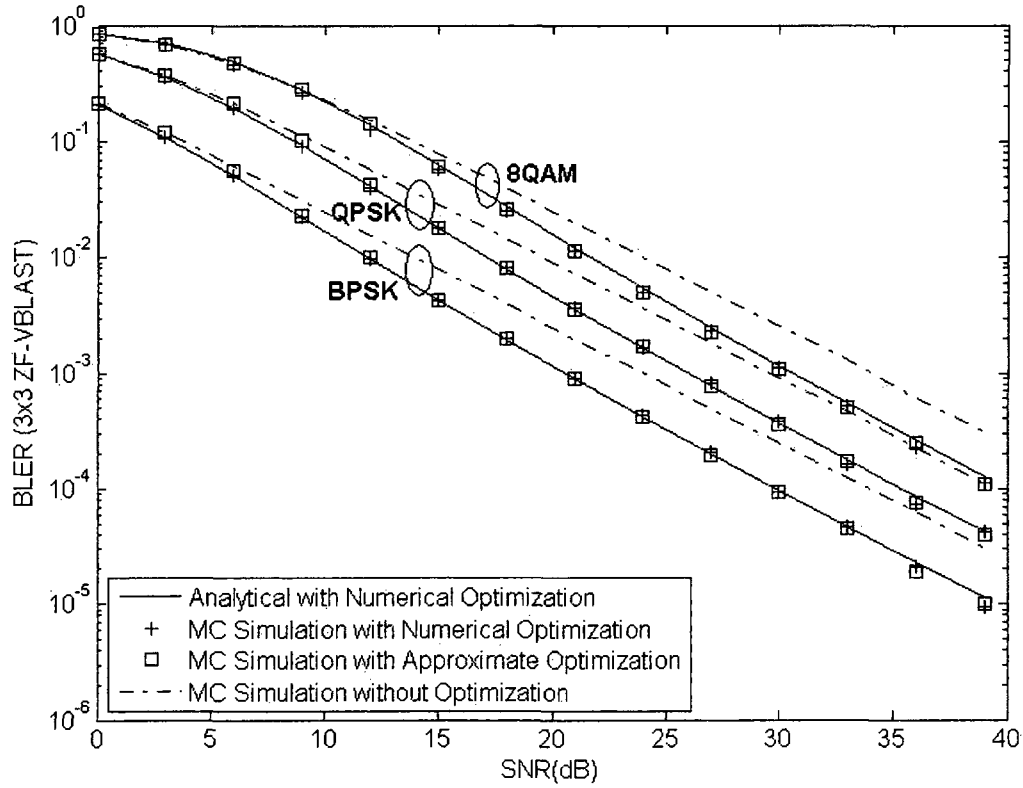


Fig. 5.2 Analytical BLER and MC simulation results of BLER with numerical and approximate α_{opt} (3×3 ZF V-BLAST) vs. SNR

2. Optimum Power Allocation for Small SNR Cases

The expression of BLER given by

$$\bar{P}_B = 1 - \prod_{i=1}^m (1 - \bar{P}_{ei}(\gamma_i))$$

still holds when γ_i is close to zero.

Figure 4.3 depicts the union bound still in agreement with the true SER when SNR is small; the simplified union bound in (5.1) is applicable for deriving the α_{opt} at small γ_0 .

To facilitate the following derivation of the optimum power allocation at small γ_0 , let's start from the truncated MacLaurin's series expansion of $\bar{P}_{ei}$, which assures us a closed-form target function $\bar{P}_B(\alpha)$. This MacLaurin's series of $\bar{P}_{ei}$ is given by

$$\bar{P}_{ei} \approx \frac{1}{2} + d_i \sqrt{\alpha_i D_s \gamma_0} \quad (5.12)$$

where $d_i = \partial \bar{P}_{ei} / \partial \sqrt{\alpha_i D_s \gamma_0}$, the derivative of $\bar{P}_{ei}$ around $\gamma_0 = 0$, equals (see Appendix A6)

$$d_i = N_c \left[-\frac{n-m+i}{2} + \frac{1}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} C_{n-m+k+i-1}^k \frac{k}{2^k} \right] \quad (5.13)$$

Hence, $\bar{P}_B$ has the approximation

$$\bar{P}_B = 1 - \prod_i (1 - \bar{P}_{ei}) \approx 1 - \prod_{i=1}^m \left(\frac{1}{2} - d_i \sqrt{\alpha_i D_s \gamma_0} \right) \approx 1 - \frac{1}{2^m} + \frac{1}{2^{m-1}} \sum_{i=1}^m d_i \sqrt{\alpha_i D_s \gamma_0}$$

Ultimately, the derivative of the Lagrange Multiplier under the power constraint $\sum_i \alpha_i = m$ about

$\bar{P}_B(\alpha_{\text{opt}})$ ends up in the form of

$$\begin{cases} L(\alpha) = \bar{P}_B(\alpha) + \lambda \left(\sum_{i=1}^m \alpha_i - m \right) \\ \left. \frac{\partial L(\alpha)}{\partial \alpha_i} \right|_{\alpha=\alpha_{\text{opt}}} = \frac{d_i}{2^m} \sqrt{\frac{D_s \gamma_0}{\alpha_{i,\text{opt}}}} + \lambda = 0 \\ \sum_i \alpha_{i,\text{opt}} = m \end{cases} \quad (5.14)$$

From the second equation, there is

$$\alpha_{i,\text{opt}} \approx \frac{d_i^2 D_s \gamma_0}{2^{2m} \lambda^2}$$

Substituting the above into the total power constraint, we get the λ and express α_{opt} :

$$\alpha_{i,\text{opt}} \approx \frac{d_i^2 m}{\sum_i d_i^2} = \frac{m \left(-\frac{n-m+i}{2} + \frac{1}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} C_{n-m+k+i-1}^k \frac{k}{2^k} \right)^2}{\sum_{i=1}^m \left(-\frac{n-m+i}{2} + \frac{1}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} C_{n-m+k+i-1}^k \frac{k}{2^k} \right)^2} \quad (5.15)$$

when

$$\lambda = \sqrt{\frac{\sum_i d_i^2 D_s \gamma_0}{m \cdot 2^{2m}}} \quad (5.16)$$

i.e., for 2×2 and 3×3 cases, the optimum power allocation coefficients around $\gamma_0 = 0$ are

Power - optimized System	Power Allocation Approximation (when γ_0 close to 0, using (5.15))
2×2 optimized ZF-VBLAST	$\alpha_1 \approx 0.6154$; $\alpha_2 \approx 1.3846$
3×3 optimized ZF-VBLAST	$\alpha_1 \approx 0.4434$; $\alpha_2 \approx 0.9977$; $\alpha_3 \approx 1.5589$

The above expression of α_{opt} justifies that α_{opt} close to $\gamma_0 = 0$ is independent of modulations. It is determined only by the number of signal transmission branches. The values in the above table agree with the numerical optimum power allocation coefficients illustrated in Figure 5.3 and Figure 5.4, where the numerical optimizations for small SNR are derived using the (SQP) function—the “fmincon” in Matlab®.

Last but not least, the analytical average BLER of the un-optimized, numerically optimized and approximately optimized ZF V-BLAST with various modulations are compared in Figure 5.5. Although Figure 5.5 shows that different modulations with the optimum power allocation may asymptotically achieve a performance improvement of $10\log(m)\text{dB}$, which is also shown in Chapter 7, higher level modulations attain less performance gain at any finite γ_0 . On the other hand, it is obvious that the optimum power allocation cannot significantly boost the performance in the small γ_0 region, although Chapter 7 shows that there is still a little SNR gain when γ_0 is

small. As a result, the inaccuracy of α_{opt} approximation in (5.15) does not cause any impairment to the performance from a practical point of view.

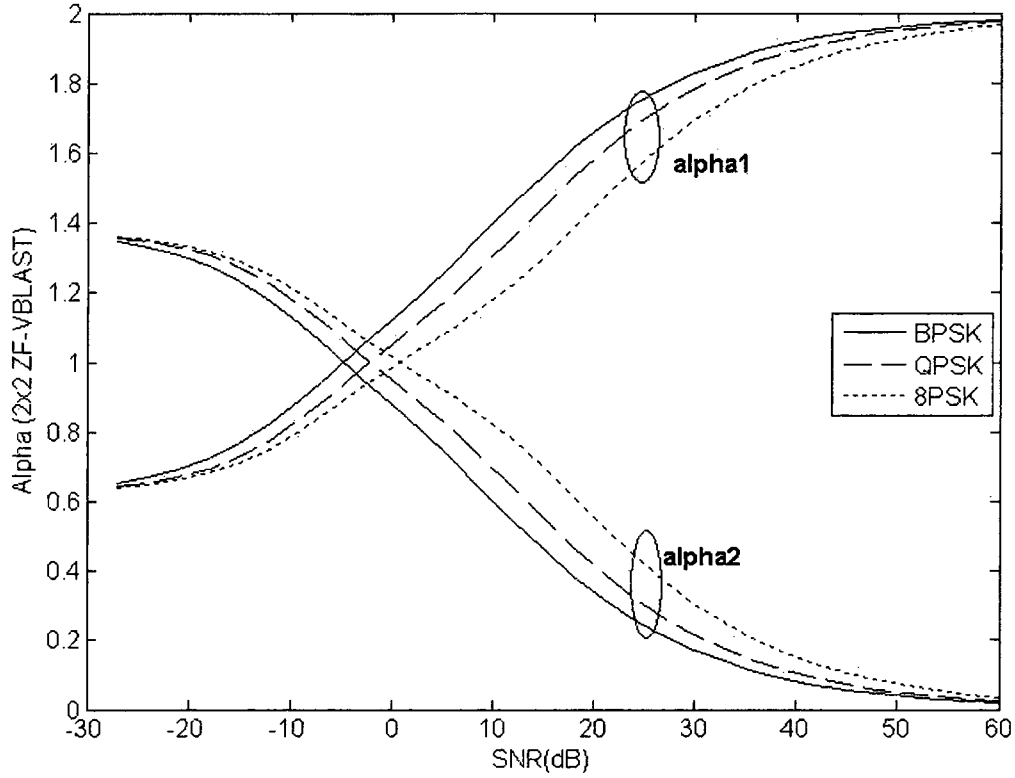


Fig. 5.3a Numerical α_{opt} for MPSK (2×2 ZF-V-BLAST) vs. SNR

5.3. Summary

[32] studied optimum power allocation strategies for BPSK based on both BLER and TBER. In [32], it was also proved that the performance improvements caused by these two strategies are almost the same. As the computationally complex error propagations should be investigated when deriving the TBER-based optimization, we focused on the general algorithm for BLER-based algorithm for different modulations here.

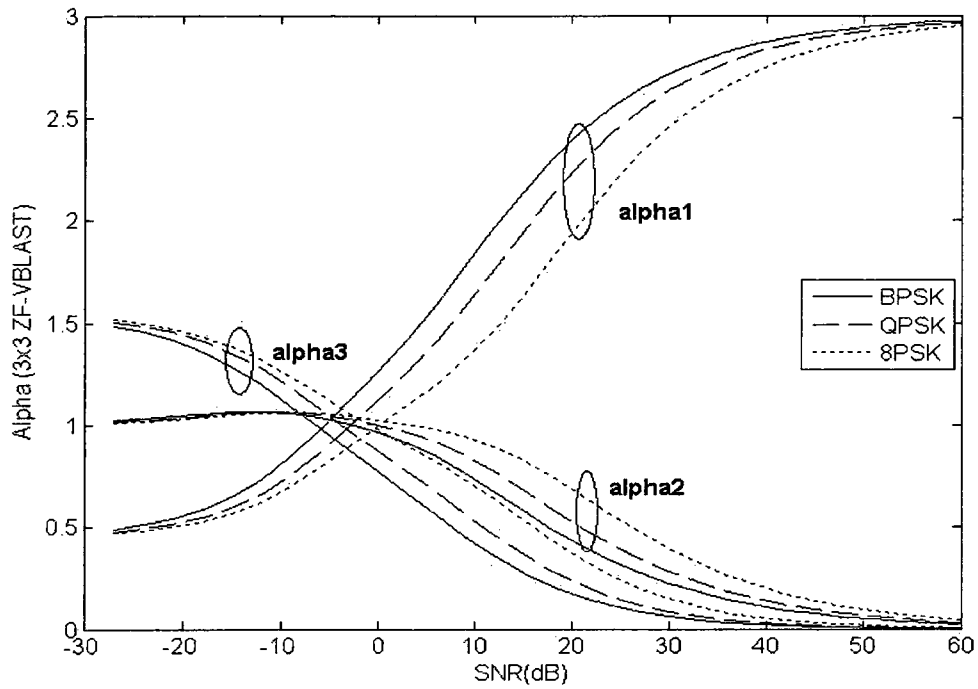


Fig. 5.3b Numerical α_{opt} for MPSK (3x3 ZF-V-BLAST) vs. SNR

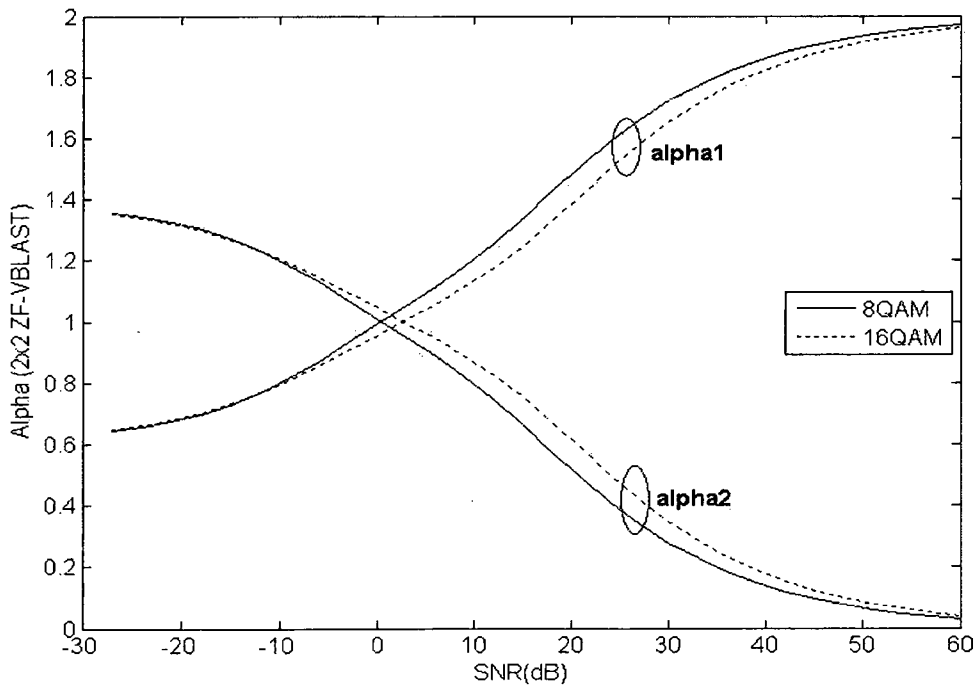


Fig. 5.4a Numerical α_{opt} for MQAM (2x2 ZF-V-BLAST) vs. SNR

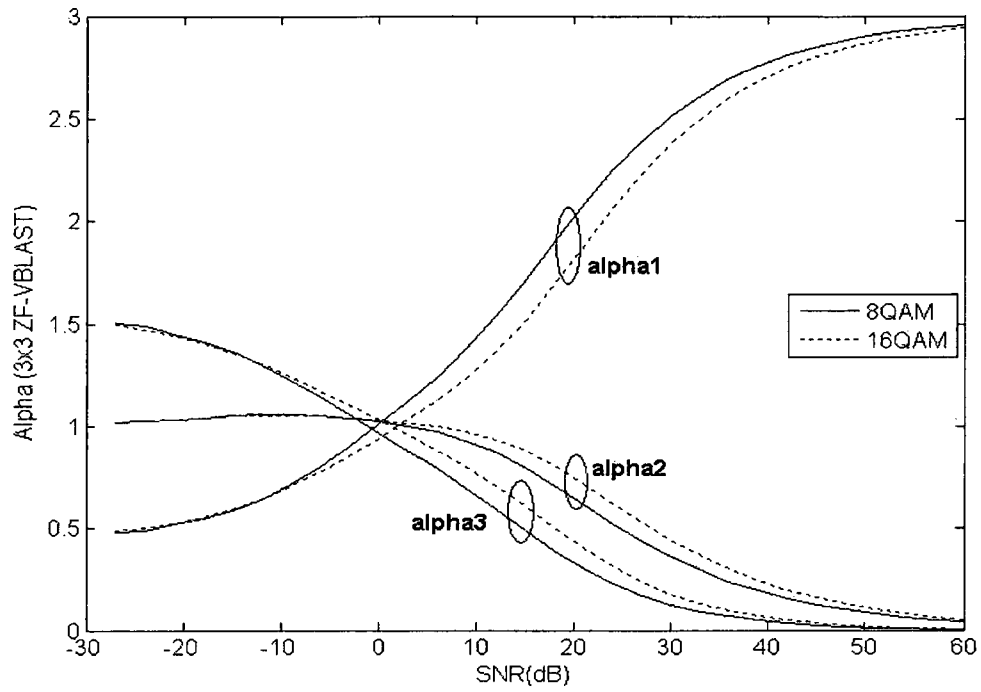


Fig. 5.3b Numerical α_{opt} for MQAM (3x3 ZF-V-BLAST) vs. SNR

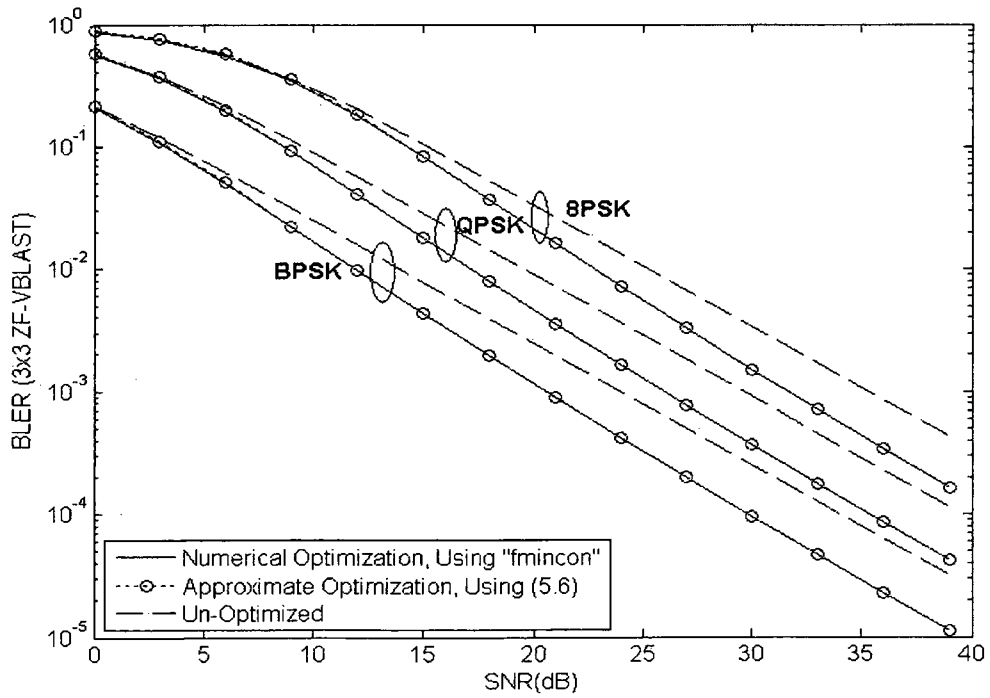


Fig. 5.5a. Analytical Average BLER of MPSK with different optimizations (3x3 ZF-VBLST)

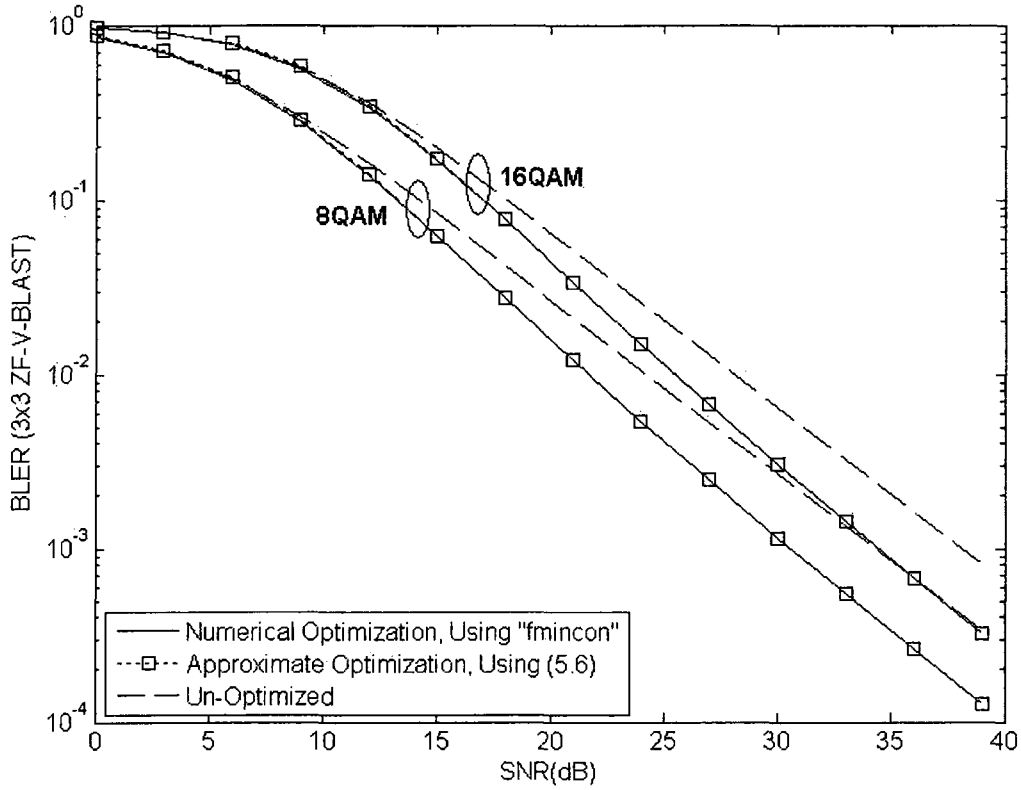


Fig. 5.5b. Analytical Average BLER of MQAM with different optimizations (3×3 ZF-VBLST) vs. SNR

The fact that the lower V-BLAST detection steps associated with the smaller diversity order suffer from a higher conditional error rate ([43]) leads to the intuitive expectation that the first step should be given most of the transmission power in an optimized algorithm. Our analysis results in this chapter definitely agree with this expectation. Note that instantaneous optimization employed by [32] requires SNR measures feedback for each channel realization, but only gains marginal performance improvement, so only the schemes based on average error rate were investigated in this chapter.

From the observations in this chapter, it can be concluded that the optimum power allocation coefficients for various modulations converge to certain numbers at both high and low

average SNR, but the convergence rate slows when the modulation level increases. Similarly, when we examine the performance improvements between the optimized and un-optimized algorithms, the average BLER of the optimized systems converges to $1/m$ of the average BLER of the un-optimized ones, whereas the higher level modulation corresponds to slower convergence.

At last but not least, the general optimization solution in this chapter makes a look-up power allocation table an alternative to the intensive optimum ordering and provides an order of magnitude of savings in the computational cost.

Chapter 6. Robustness of the Optimum Allocation

In real wireless systems, receivers cannot assure perfect channel knowledge for coherent detections. It is conceivable that inaccurate estimations of $\mathbf{H}$ and γ_0 harm optimum power allocation. From a practical perspective, a “good” algorithm must be “robust” and the performance degeneration should be acceptable when the abovementioned inaccuracies occur. Conversely, the algorithm is “sensitive” and “unstable” if a small inaccuracy leads to significant performance variation. For our optimum power allocation, disturbances in α_{opt} due to channel estimation inaccuracy pose a threat to the optimization. Thus, examining the sensitivity of $\bar{P}_B(\alpha)$ to α around α_{opt} is necessary. The fluctuations of α_{opt} are attributed to either the total transmit power $\sum_i \alpha_i = u$ variation (i.e., the perturbation of amplifier outputs) or the approximations in the algorithm. For these reasons, a sensitivity measurement δ is suggested by

$$\delta = \left| \frac{\Delta \bar{P}_B / \bar{P}_B}{\Delta u / u} \right| \quad (6.1)$$

or

$$\delta = \left| \frac{\Delta \bar{P}_B / \bar{P}_B}{\Delta \alpha_i / \alpha_i} \right| \quad (6.2)$$

If the fluctuation is small, the above definitions can be approximated further in the forms of the derivative

$$\frac{\Delta \bar{P}_B / \bar{P}_B}{\Delta u / u} \approx \frac{\partial \bar{P}_B}{\partial u} \frac{u}{\bar{P}_B} \quad (6.3)$$

or

$$\frac{\Delta \bar{P}_B / \bar{P}_B}{\Delta \alpha_i / \alpha_i} \approx \frac{\partial \bar{P}_B}{\partial \alpha_i} \frac{\alpha_i}{\bar{P}_B} \quad (6.4)$$

6.1. Local Robustness

The derivative of $\overline{P_B}$ over α_i , $\partial \overline{P_B}(\alpha) / \partial \alpha_i$ can be found at the preceding Lagrange equation (5.5).

$$\left. \frac{\partial L(\alpha)}{\partial \alpha_i} \right|_{\alpha_i = \alpha_{i,\text{opt}}} = \left. \frac{\partial \overline{P_B}(\alpha)}{\partial \alpha_i} \right|_{\alpha_i = \alpha_{i,\text{opt}}} + \lambda = 0 \quad (6.5)$$

when the above Lagrange equation holds, $\partial \overline{P_B}(\alpha) / \partial \alpha_i$ equals $-\lambda$. Since $\partial \overline{P_B} / \partial u = \sum_i \left[(\partial \overline{P_B} / \partial \alpha_i) (\partial \alpha_i / \partial u) \right] = -\lambda \sum_i (\partial \alpha_i / \partial u) = -\lambda$ too, when the fluctuation of u is limited to a small range around α_{opt} , the $\partial \overline{P_B} / \partial u$ can be merged into the measure δ_i' , “local robustness,” as:

$$\delta_i' = \frac{\partial \overline{P_B} / \overline{P_B}}{\partial \alpha_i / \alpha_i} = \left. \frac{\partial \overline{P_B}}{\partial \alpha_i} \right|_{\alpha_i = \alpha_{i,\text{opt}}} \cdot \frac{\alpha_{i,\text{opt}}}{\overline{P_B}} = \lambda \frac{\alpha_{i,\text{opt}}}{\overline{P_B}} \quad (6.6)$$

where the Lagrange multiplier λ is defined in (5.7). In Appendix A7, λ is approximated as

$$\lambda \approx \left[N_c (n - m + 1) / m \right] \left[C_{2(n-m+1)-1}^{n-m+1} / (4D_s m \gamma_0)^{n-m+1} \right] \quad (6.7)$$

$\delta_i'(\alpha_{i,\text{opt}})$ is then proved to be

$$\delta_i' = \lambda \frac{\alpha_{i,\text{opt}}}{\overline{P_B}} \approx \frac{(n - m + 1)}{m(4D_s \gamma_0)^{\frac{i-1}{n-m+i+1}}} \sqrt[n-m+i+1]{\frac{(n - m + i) C_{2(n-m+i)-1}^{n-m+i} \cdot m^{n-m+2}}{(n - m + 1) C_{2(n-m+1)-1}^{n-m+1}}} \quad (6.8)$$

i.e., when $i = 1$, $\delta_1' \approx n - m + 1$, which equals the diversity order of this step detection. In addition, when $n = m$, compare to the other scenarios with $n > m$, the sensitivity of the first step at $\alpha_{1,\text{opt}}$ is minimum; e.g., $\delta_1' \approx 1$, which is an ideal condition $\Delta \overline{P_B} / \overline{P_B} \approx \Delta \alpha_i / \alpha_i$; i.e., the normalized variation of the BLER equals to the normalized variation of the power allocation. It should be emphasized that since $\gamma_0 \gg 1$, the superiority, in terms of robustness, of the higher steps ($i > 1$) over the first step is obvious.

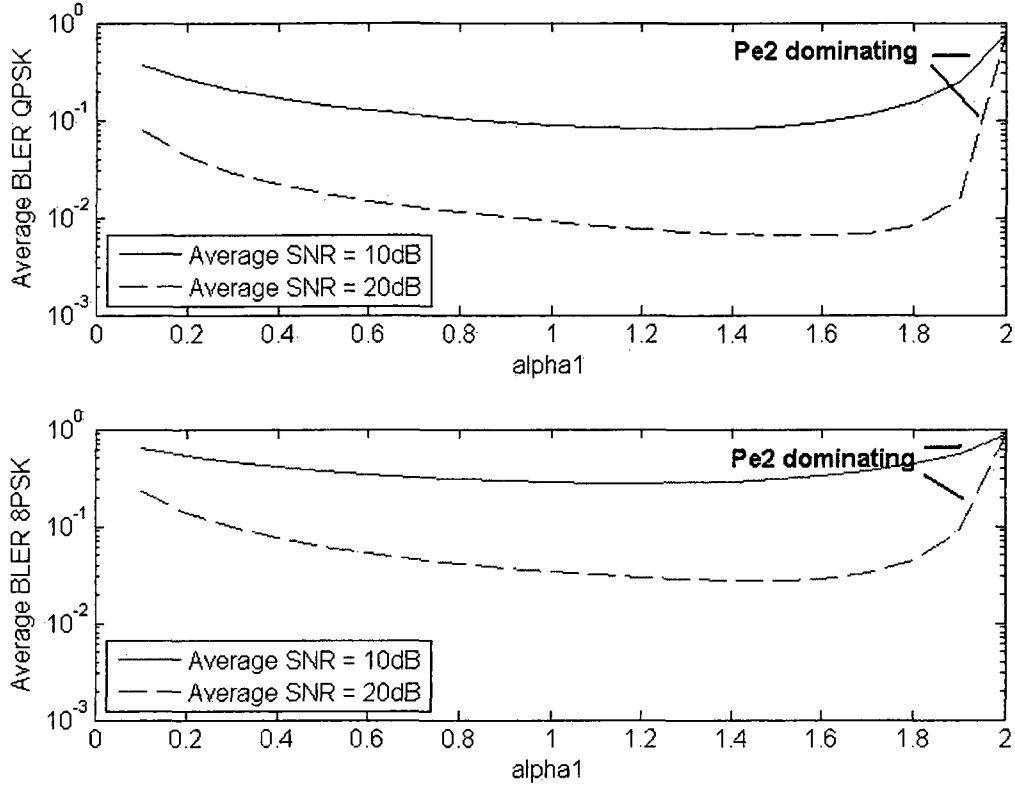


Fig. 6.1 Average BLER vs. α_1 (2×2 ZF-V-BLAST with QPSK and 8PSK)

In the Figure 6.1, $\bar{P}_B(\alpha)$ curves around $\alpha_{1,opt}$ are flat, which means that the average BLER is not sensitive to α_1 perturbation around $\alpha_{1,opt}$. The heuristic observations on above curves include:

- First, the higher level modulations associated with smaller D_s lead to the larger δ_i' and δ_i' is independent of the parameter N_c . However, the difference of δ_i' between two modulations caused by D_s is extremely small; e.g., $\delta_{2,16QAM}' = \sqrt[3]{2} \cdot \delta_{2,8QAM}'$ given $D_{s,16QAM} = 0.5 \cdot D_{s,8QAM}$ (Table.4.1) in the 2nd step of a 2×2 system.
- Second, of course, when α_1 is far away from $\alpha_{1,opt}$ (i.e., too much power is allocated to the first transmitter), the slope of $\bar{P}_B(\alpha)$ becomes steep in the “ $\bar{P}_{e2}$ dominating” area [32]. As the

curves of different modulations converge to one (i.e., zero power on the second branch results in 100% BLER), the curves of lower level modulations are usually associated with a steeper slope when α_1 is close to m .

- Third, as a result, even for the high-level modulations, the optimum algorithm exhibits robustness at the higher steps as long as n is close to m .
- Ultimately, local robustness decreases when the efforts are made to increase the diversity order $(n - m + i)$ ([32]).

6.2. Global Robustness

When the global robustness is examined, the variation of power allocation is not necessarily small around $\alpha_{i,\text{opt}}$. With the general theory of the convex optimization ([8]), the following inequality holds

$$\Delta \overline{P}_B = \overline{P}_B(\mathbf{\alpha}') - \overline{P}_B(\mathbf{\alpha}_{\text{opt}}) \geq -\lambda \cdot \Delta \alpha_i \quad (6.9)$$

for single branch power variation, where $\mathbf{\alpha}' = [\alpha_{1,\text{opt}}, \alpha_{2,\text{opt}}, \dots, \alpha_{i,\text{opt}} + \Delta \alpha_i, \dots, \alpha_{m,\text{opt}}]$ and $\lambda > 0$.

Normalizing both sides of the inequality, we obtain

$$\left| \frac{\Delta \overline{P}_B}{\overline{P}_B(\mathbf{\alpha}_{\text{opt}})} \right| \bigg/ \frac{\Delta \alpha_i}{\alpha_{i,\text{opt}}} \leq \delta', (\Delta \alpha_i > 0); \frac{\Delta \overline{P}_B}{\overline{P}_B(\mathbf{\alpha}_{\text{opt}})} \bigg/ \left| \frac{\Delta \alpha_i}{\alpha_{i,\text{opt}}} \right| \geq \delta', (\Delta \alpha_i < 0) \quad (6.10)$$

Therefore, the “local sensitivity” δ' in (6.8) also serves as the upper or lower bound of the “global sensitivity.” Since the average BLER increases by at least $\delta' \overline{P}_B(\mathbf{\alpha}_{\text{opt}}) |\Delta \alpha_i| / \alpha_{i,\text{opt}}$, the algorithm could be sensitive when the extremely negative $\Delta \alpha_i$ occurs.

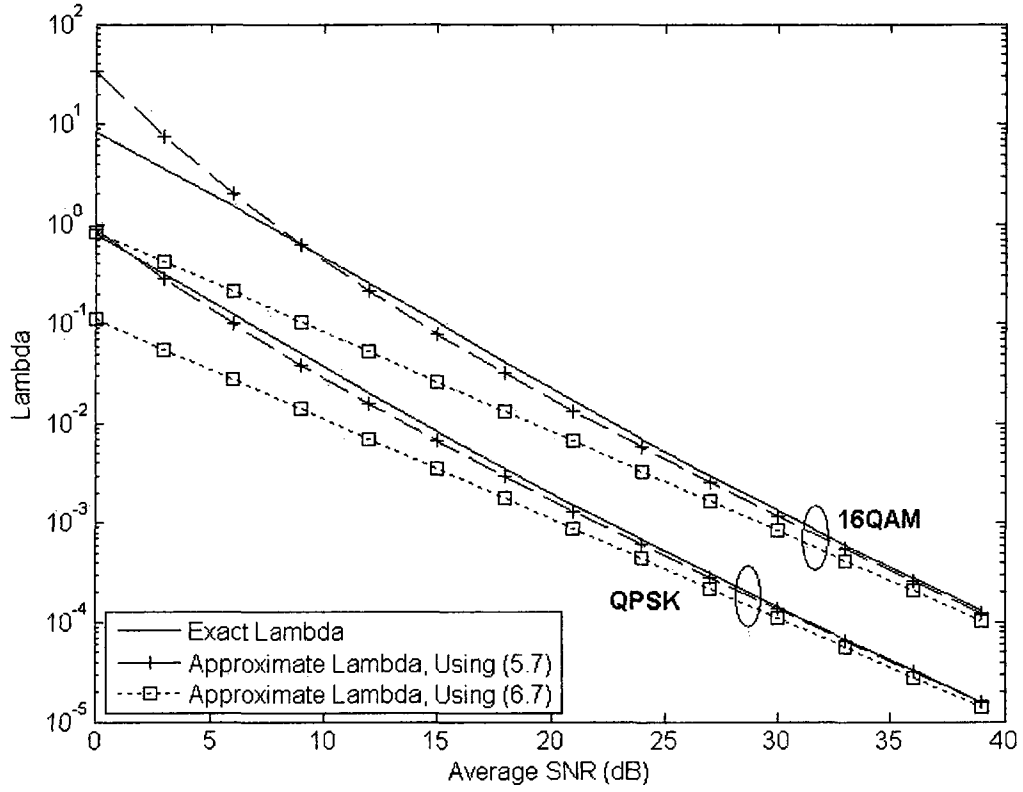


Fig. 6.2 Exact and approximate Lagrange multiplier λ vs. average SNR (3×3 ZF-VBLAST with QPSK and 16QAM)

Since the Lagrange multiplier λ is derived when $\Delta\alpha_i$ is close to zero in an asymptotic case, this global sensitivity is lower bounded by the local sensitivity asymptotically as well. In Figure 6.2, the numerical $\alpha_{i,\text{opt}}$ is used to attain the “exact” λ ; it can be claimed that the approximate λ in (6.7) converges to the exact λ asymptotically. Notably, the convergence of both approximations in

$$\begin{cases} \lambda \approx \frac{1}{(4D_s\gamma_0)^{(n-m+1)}} \left(\frac{a_1}{m}\right)^{n-m+2} \left[1 - \frac{b_2}{mc_1^{n-m+3}\sqrt{4D_s\gamma_0}}\right]^{-(n-m+2)c_1} \\ a_1 = {}^{n-m+2}\sqrt{N_c(n-m+1)C_{2(n-m+1)-1}^{n-m+1}}; b_2 = {}^{n-m+3}\sqrt{\frac{(n-m+2)C_{2(n-m+2)-1}^{n-m+2} \cdot m^{n-m+2}}{(n-m+1)C_{2(n-m+1)-1}^{n-m+1}}} \\ c_1 = \frac{(n+1)!}{(n-m+1)!(n-m+2)} \end{cases}$$

and (6.7) slow down when the modulation level increases (i.e., A decreases.) Since the approximation in (5.7) is more accurate, one may conclude that λ approximation (5.7) is applicable to the “global sensitivity” derivation for high-level modulations, especially at moderate γ_0 .

6.3. Summary

In this chapter, we examined the sensitivity of BLER in terms of power allocation variations. Two measures of this sensitivity (i.e., the “local” and “global” robustness quantified by the abovementioned Lagrange optimization multiplier) are introduced. Two different power variations—single transmitter power and total transmitter power—are merged in the robustness measures. Although the examples in Figure 6.1 show that the algorithm becomes sensitive when the power allocation deviates too much from the optimum value and the lower-level modulations suffer when the slopes are steeper, the robustness in the wide range around the optimum $\alpha_{i,\text{opt}}$ still exemplifies that in practical respect, if an average power allocation is not feasible, a pre-set power allocation $\alpha_{\text{opt}} = [\alpha_{1,\text{opt}}(A, \overline{\gamma_0}), \alpha_{2,\text{opt}}(A, \overline{\gamma_0}), \dots, \alpha_{m,\text{opt}}(A, \overline{\gamma_0})]$ based on the average SNR $\overline{\gamma_0}$ over a period of time is promising.

Chapter 7. SNR Gain

In order to quantify the performance improvement achieved through the use of optimized non-uniform power allocation, in this chapter we define a performance measure, the SNR gain of the optimized system, as

$$G_{\text{SNR}} = \alpha$$

when the BLER of the optimum power allocated system has a performance of

$$\overline{P}_B(\alpha_{\text{opt}}) = \overline{P}_B(\alpha, \alpha, \dots, \alpha) \quad (7.1)$$

The right-hand side of this equation represents the performance of the system with uniform power allocation $\alpha = [\alpha, \alpha, \dots, \alpha]$. Analytical approximations of this SNR gain in the large and small SNR cases are derived. Both these approximations are general expressions parameterized with normalized minimum distance in the constellations A for various modulations.

7.1. General Bound of SNR Gain

By the definition of SNR gain in (7.1), the intuitive expectation is that the gain is bounded by

$$1 \leq G_{\text{SNR}} \leq m \quad (7.2)$$

because it is conceivable for the optimized system to outperform the non-optimized system $\overline{P}_B(1, 1, \dots, 1)$ but be inferior to the system with higher transmit power $\overline{P}_B(m, m, \dots, m)$ due to the constrain $\sum_i \alpha_{i,\text{opt}} = u = m$, i.e., $\alpha_{i,\text{opt}} \leq m$. Note that it follows from $\overline{P}_B(1, 1, \dots, 1) \leq \overline{P}_B(\alpha_{\text{opt}}) \leq \overline{P}_B(m, m, \dots, m)$, where the asymptotical approximation of $\overline{P}_B(\alpha_{\text{opt}})$ is the summation of a monotonically decreasing function $\overline{P}_{ei}(\alpha_i)$.

7.2. SNR Gain Asymptotic Behavior

The asymptotic approximation of $\bar{P}_B(\mathbf{a})$ is given by (5.3),

$$\bar{P}_B(\mathbf{a}_{\text{opt}}) \approx N_c \sum_{i=1}^m \frac{C_{2(n-m+i)-1}^{n-m+i}}{(4\alpha_{i,\text{opt}} \cdot D_s \gamma_0)^{n-m+i}}$$

Since the BLER is dominated by the $\bar{P}_{e,1}$, $\bar{P}_B$ may be further approximated by keeping the first term of the above summation as (see Appendix.A8)

$$\begin{aligned} \sum_{i=1}^m \bar{P}_{e,i}(\mathbf{a}_{\text{opt}}) &\approx \bar{P}_{e,1}(\alpha_{1,\text{opt}}) \approx N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{(4D_s \gamma_0 m)^{n-m+1}} \cdot \left[1 + \frac{n-m+1}{m} b_2 (4D_s \gamma_0)^{-\frac{1}{n-m+3}} \right] \\ b_2 &= \sqrt[n-m+3]{\frac{(n-m+2)C_{2(n-m+2)-1}^{n-m+2} \cdot m^{n-m+2}}{(n-m+1)C_{2(n-m+1)-1}^{n-m+1}}} \end{aligned} \quad (7.3)$$

On the other hand, for the $\bar{P}_B$ with uniform power allocation $\mathbf{a} = [\alpha, \alpha, \dots, \alpha]$ is approximately expressed as

$$\sum_{i=1}^m \bar{P}_e(\alpha) \approx \bar{P}_{e,1}(\alpha) \approx N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{(4\alpha D_s \gamma_0)^{n-m+1}} \quad (7.4)$$

To make (7.3) equals (7.4), α as the SNR gain should equal

$$\alpha = G_{\text{SNR}} \approx \frac{m}{\sqrt[n-m+1]{1 + \frac{(n-m+1)b_2}{m(4D_s \gamma_0)^{-\frac{1}{n-m+3}}}}} \quad (7.5)$$

Therefore, G_{SNR} is independent of N_c but depends on D_s .

In Figure 7.1, the two numerical G_{SNR} 's are obtained from the numerically derived $\mathbf{a}_{\text{opt}}$ in Figure 5.1a and the approximate $\mathbf{a}_{\text{opt}}$ in (5.6). The abovementioned closed form gain from (7.5) is also given for comparison purpose. Here, the Matlab® SQP subroutine “fsolve” is employed to numerically find the “numerical” G_{SNR} —the solution of the nonlinear equation (7.1). According to Figure 7.1 and also from (7.5), it is obvious that $G_{\text{SNR}} \rightarrow m$ when $\gamma_0 \rightarrow \infty$. That is because α_i

$\rightarrow m$ as $\gamma_0 \rightarrow \infty$ and then the first step SER $\bar{P}_{e,1}(m)$ dominates the average BLER. As for the observations of modulation-dependent behaviors, the convergence of $G_{\text{SNR}} \rightarrow m$ in higher level modulations is slower than that in lower level modulations. This observation is explained by the fact that convergence $n-m+3\sqrt{4D_s\gamma_0} \gg 1$ in a higher level modulation is slowed by the smaller D_s .

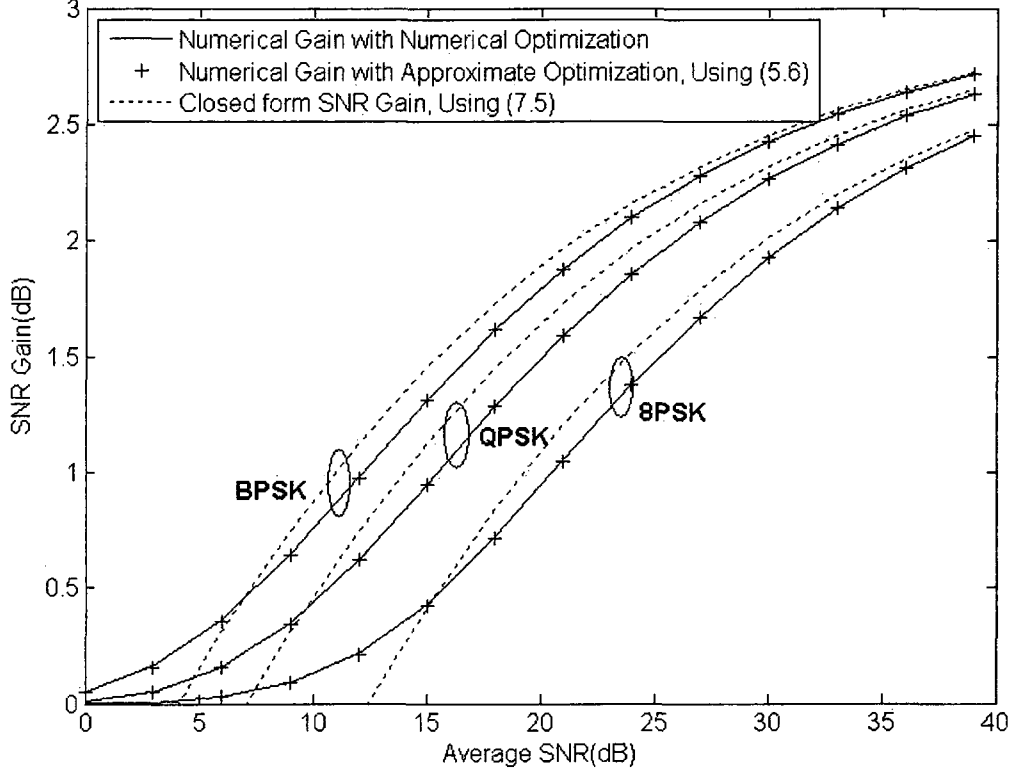


Fig. 7.1 Numerical and closed form SNR gains (2x2 ZF-VBLAST with MPSK) vs. SNR

7.3. SNR Gain Behavior at Small SNR Region

Instead of being approximated by (5.3), for the small SNR cases, the SNR gain is close to

$$\bar{P}_B = 1 - \prod_i \bar{P}_{ei} \approx 1 - \prod_{i=1}^m \left(\frac{1}{2} + d_i \sqrt{\alpha_i D_s \gamma_0} \right) \approx 1 - \frac{1}{2^m} + \frac{1}{2^{m-1}} \sum_{i=1}^m d_i \sqrt{\alpha_i D_s \gamma_0}$$

In addition, because the α_{opt} for small SNR is given by (5.15), the above approximation is finally equals to

$$\begin{aligned}\overline{P}_B(\alpha_{\text{opt}}) &\approx 1 - 2^{-m} + 2^{-m+1} \sqrt{mD_s\gamma_0} \sum_{i=1}^m d_i \sqrt{\frac{d_i^2}{\sum_i d_i^2}} \\ &= 1 - 2^{-m} + 2^{-m+1} \sqrt{mD_s\gamma_0} \sqrt{\sum_i d_i^2}\end{aligned}\quad (7.6)$$

On the other hand, if the α is uniformly allocated as $[\alpha, \alpha, \dots, \alpha]$, $\overline{P}_B$ at small SNR is in the form of

$$\overline{P}_B(\alpha, \alpha, \dots, \alpha) \approx 1 - 2^{-m} + 2^{-m+1} \sqrt{\alpha D_s \gamma_0} \sum_{i=1}^m d_i \quad (7.7)$$

Thus, according to the definition of the SNR gain,

$$\alpha = G_{0,\text{SNR}} \approx \frac{m \sum_{i=1}^m d_i^2}{\left(\sum_{i=1}^m d_i\right)^2} \approx \frac{m \sum_{i=1}^m \left(-\frac{n-m+i}{2} + \frac{1}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} C_{n-m+k+i-1}^k \frac{k}{2^k} \right)^2}{\left[\sum_{i=1}^m \left(-\frac{n-m+i}{2} + \frac{1}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} C_{n-m+k+i-1}^k \frac{k}{2^k} \right) \right]^2} \quad (7.8)$$

e.g., $G_{0,\text{SNR}} = 1.0604$ for a 3×3 system and $G_{0,\text{SNR}} = 1.0400$ for a 2×2 system. Therefore, by observing the expression (7.8), we conclude that the approximate SNR gain behavior close to $\gamma_0 = 0$ is a function that depends only on the number of transmitter and receiver antennas (i.e., the diversity order) and is independent of the modulation schemes.

7.4. Summary

With the normalized minimum Euclidean distance $d_{\min}^*$, a constant which depends on the constellation, the asymptotic approximation of the average BLER-based SNR gain is obtained for various modulations. A closed-form low-SNR approximation of this gain was also obtained. This function depends on the number of transmitter and receiver antennas, but not on modulation schemes.

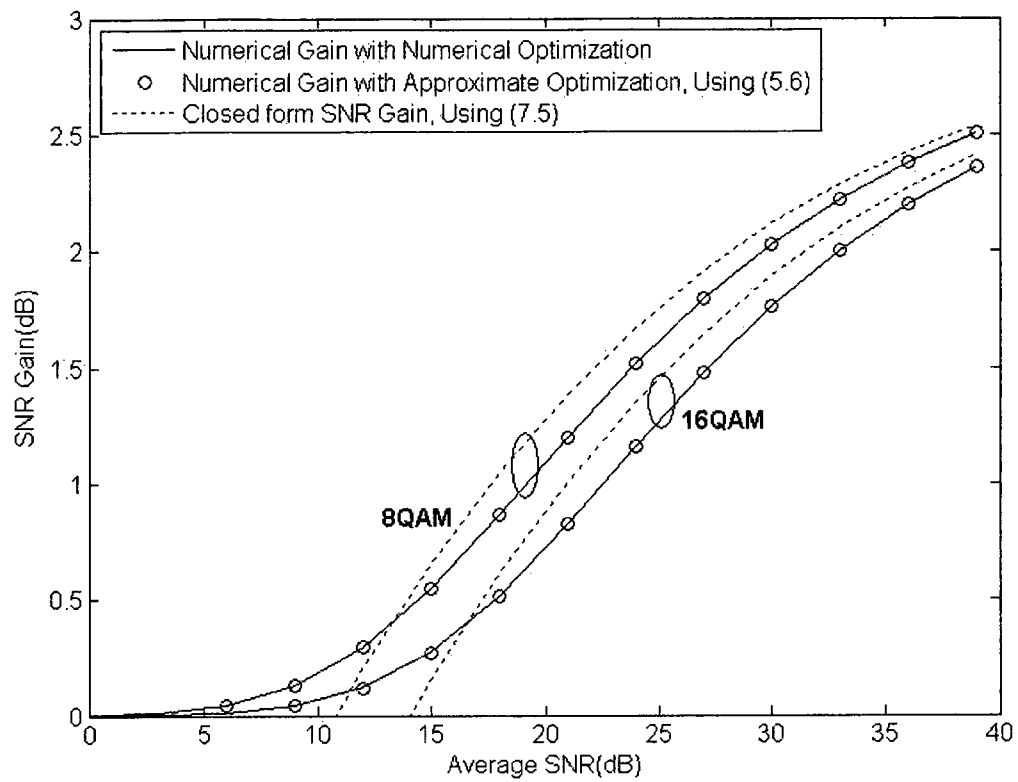


Fig. 7.2 Numerical and closed form SNR gains (2×2 ZF-VBLAST with MQAM) vs. SNR

Chapter 8. Parallel MC Simulation

There are currently quite a few toolkits to help Matlab® users solve computationally demanding problems using parallel calculations in a multi-processor or computer cluster environment. Thus, to speed up the Monte Carlo (MC) simulations in this thesis, we implement the abovementioned state-of-the-art parallel Matlab® for our optimized V-BLAST simulations for the different modulations. Among the candidate toolkits, “pMatlab,” from Massachusetts Institute of Technology (MIT), and the Parallel Computing Toolbox™ (PCT), from Mathworks™, are typical and widely accepted. Furthermore, after careful comparison, we finally picked pMatlab to implement our MC simulations due to the reasons being specified in this chapter. We also briefly review the main parallelism schemes, then evaluate pMatlab and PCT, and finally, summarize important observations regarding parallelism efficiency.

8.1. Parallelism in Matlab® Application

The current available tools for parallel Matlab® computing are categorized into three types: 1) embarrassingly parallel, 2) message-passing, and 3) backend support. A brief review of these three categories is introduced by [13]:

1. *Embarrassingly Parallel*

For this type of parallelism, the main part of the code is run on the main processor, while the “sub-processes,” which are usually defined as independent functions in Matlab®, are distributed among other local or remote processors. When all the results of the sub-processes are ready and have been collected, the rest of the main processes continue. The embarrassingly

parallel toolkits help to distribute the sub-processes' inputs and collect the corresponding outputs (Figure 8.1). The limitations of this parallelism include the following:

- Data transmissions among sub-processes are not supported;
- Iterations where data is shared across the parallel calculations are intractable; and
- Parallel calculations are uncontrollable as soon as they are started.

The advantages are that this toolkit is usually compact, and modifying the original Matlab® codes into the “parallelized” codes is usually succinct. For these reasons, both the toolkits adopted in our parallel simulation practices are actually embarrassingly parallelism types. In fact, as a MC simulation is essentially a collection of independent experiments with internal random variable generations, the data communication and main process intervention are not required in our simulation cases.

2. Message-Passing

The main difference between message-passing and embarrassingly parallel parallelism is that in message-passing parallelism, every single sub-process running on separate processors can be invoked or manipulated by peer sub-processes or the main process (Figure 8.1). Although the data structure in this parallel becomes very complex, the advantages are obvious:

- Operations on sub-processes become very flexible; and
- The compiled parallel commands offer syntax similar to the original Matlab® commands, and the parallelism is transparent to the code developer.

The bottleneck of this parallelism arises from its demanding nature: The overhead data preparation and transmission is costly, especially regarding calculation efficiency. Moreover,

most of today's toolkits offer only limited compiled parallel replacements for the conventional Matlab® functions. Finally, robustness is penalized in this type of parallelism.

3. Backend Support

This scheme is not a type of parallelism in the strict sense; rather, a “distributing/gathering” architecture had been widely used in laboratory environment before the multi-core computer was introduced. In Figure 8.1, the front ends only work as the user interfaces pack and send tasks to the backend, a powerful computation engine where the calculations are performed and the results are returned to the front ends automatically or upon request.

This “parallelism” is capped by an unknown waiting time in the queue, the costly calculation engines and the low efficiency of data transmission.

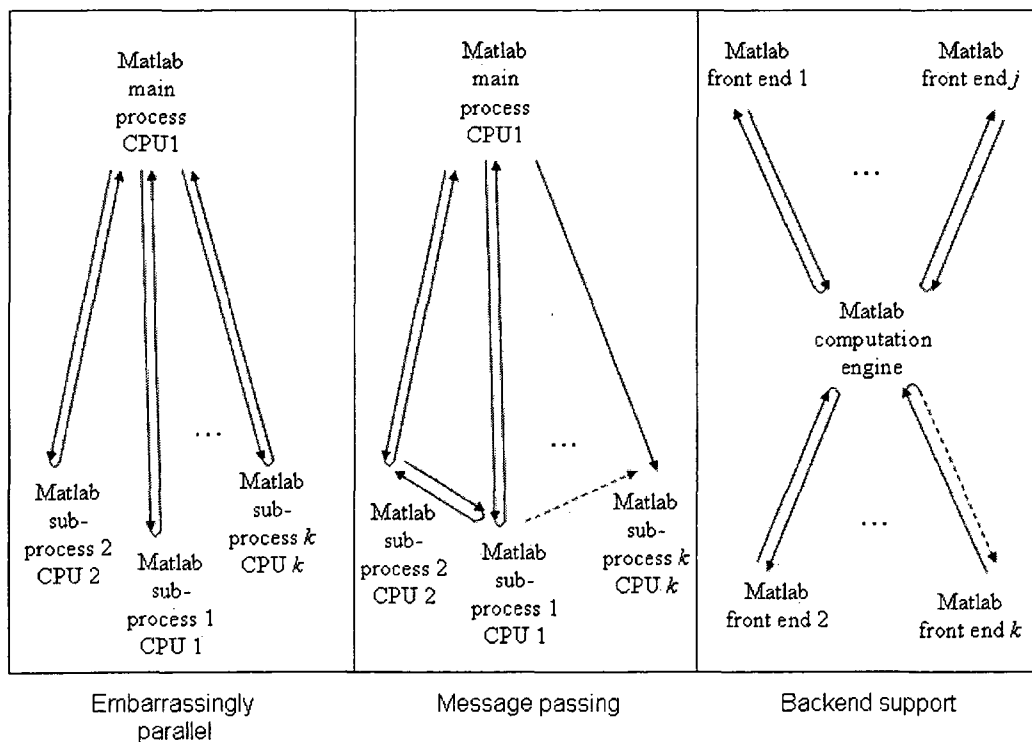


Fig. 8.1 Three types of parallelism (according to [13])

When we screened the existing toolkits (Appendix.B1), we found that most are not compatible with a PC/Windows® environment, on which our simulation implementations are based. Some were being developed to support the new Windows® operating systems, but this development has been suspended; some previously free toolkits now carry a price; and others require an UNIX environment to support workload sharing.

The candidate list was then narrowed to a choice between MIT's "pMatlab" and Mathwork's PCT.

pMatlab	PCT
Free tool package and easy installation	Commercial software, time-limited license and professional assistant required during the installation
Compatible with any existing Matlab® version	Additional "Matlab Distributed Computing Engine" (MDCE) from Mathworks™ required to identify the distribution environment for versions prior to R2007b
Pre-defined distributed arrays, according to the number of CPU "workers," is demanded	Pre-defined distributed arrays are not necessary
Messy "parallelizing" coding syntax	Straightforward operation in "pmode" and easy "parallelizing" recoding
Inaccessible "my_results" local sections disadvantageous for debugging practices	Accessible local sections of the distributed array
Limited message-passing functions	Comprehensive message-passing functions and most existing Matlab® functions supported

Table. 8.1 Comparison between toolkits pMatlab and PCT

Ultimately, as pMatlab—a third-party package based on the Message-Passing Interface (MPI) specification⁴—is free, we used pMatlab to implement the following MC applications in parallel. (Refer to Appendix.C3 to learn how to “parallelize” Matlab® simulation codes to those pMatlab can recognize.) On the other hand, in a 30-day trial, PCT™, as a commercial toolkit developed by Mathwork™, outperformed pMatlab with complete embarrassingly and message-passing parallelism⁵ features, universal user interface and better documentary and technical support. (See Appendix.B2 and Appendix.B3 for details on pMatlab and PCT usage.) A comparison between pMatlab and PCT is summarized in Table.8.1.

8.2. Evaluation of Parallelism Efficiency

We obtained interesting results during our parallel simulation implementations, which led to a couple of the studies shown below. Note that the factors particularized in these study conclusions essentially affect the parallel MC simulation efficiency.

1. *Analysis of pMatlab Run*

When the running time of the “serial” MC simulations is measured, it may increase linearly with the number of MC iterations. But for parallel MC simulations, it emerges that there is an approximate 10-second “initiation time” during which pMatlab prepares distributed arrays. The preparation work includes writing and buffering input and output arrays to/from the hard disk. In fact, disassembling typical pMatlab scripts—`'eval (MPI_RUN ('pMCBPSKrun', 2, {}));'`—

⁴ MPI is a communication protocol independent of coding language for parallel computation coding. Therefore, MPI works as a backend message-passing interface for communication among processes. (www.wikipedia.com)

⁵ From a data structure point of view, Mathworks™ calls embarrassingly parallel “task parallel” and message-passing “data parallel,” because the former works on the computations organized into independent processing tasks. In general cases, inputs are different from task to task and there is no communication among tasks. The latter computation bases its efforts on distributing associated data vectors to different CPUs or cluster computers. Each part of the data is processed on different hardware simultaneously. (www.mathworks.com)

reveals the command executing sequence shown in Figure 8.2. The run-time measurements in Figure 8.3 also illustrate the influence of the above initiation on the total simulation time. Generally, the initialization effect vanishes as the iteration number increases and is inconsiderable in a long-run simulation.

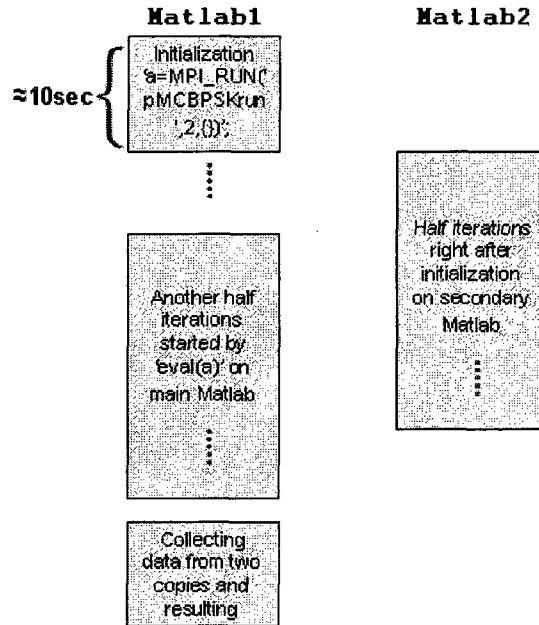


Fig. 8.2 Disassembling pMatlab process

2. Hyper threading

Another valuable investigation revealed that the “hyper threading” system allocating tasks between processors has a great deal of influence on parallel implementation efficiency. Time measurements of certain MC simulation examples (shown by Table.8.2 and Figure 8.3) and dozens of other simulation implementations corroborate that parallel efforts actually slow down when the hyper threading features of the processors are on. In Figure 8.3, the parallel simulation without hyper threading demonstrates good efficiency when it is needed (i.e., almost half cut of the original running time is achieved when the number of iteration is large.) It should be

emphasized that we don't need to implement parallel simulations when the iteration numbers and the running time are small, so low efficiency region in Figure 8.3 does not influence practical parallel implementations actually. The Similar observations were made when either pMatlab or PCT™ was used. Given the premise that each physical processor is tied up under the MPI protocol control, any “hyper threading” attempt to execute multiple concurrent Matlab® processes from the operating system perspective merely starves the limited resources and disturbs the existing processes. As a result, it is suggested that the hyper threading of each parallel Matlab® process be manually blocked in the system's “task manager” window before a parallel simulation is run. (Even though this observation has not been verified in other numerical computing environments, care should be taken when making parallel computations.)

MC Iteration No.	1.E+05	1.E+06	1.E+07	1.E+08	1.E+09
Running time in serial (s)	3.00	29.62	294.76	2952.27	30115.01
Parallel running time w/ hyper threading(s)	11.35	26.83	188.28	1815.14	18233.28
Parallel (w/ hyper threading) / Serial (%)	378.2%	90.6%	63.9%	61.5%	60.5%
Parallel running time w/o hyper threading(s)	10.85	25.81	161.00	1534.84	15405.43
Parallel (w/o hyper threading) / Serial (%)	361.7%	87.1%	54.6%	52.0%	51.2%

Table 8.2 Running time of serial Matlab vs. in pMatlab (with (w/) and without (w/o) hyper threading)

The final—but not least significant—issue that should be addressed regarding parallel Matlab® applications is that the size of the distributed arrays should not grow with the MC iterations, or the efficiency will be impaired due to the demanding data transmission between processor buffers and disks.

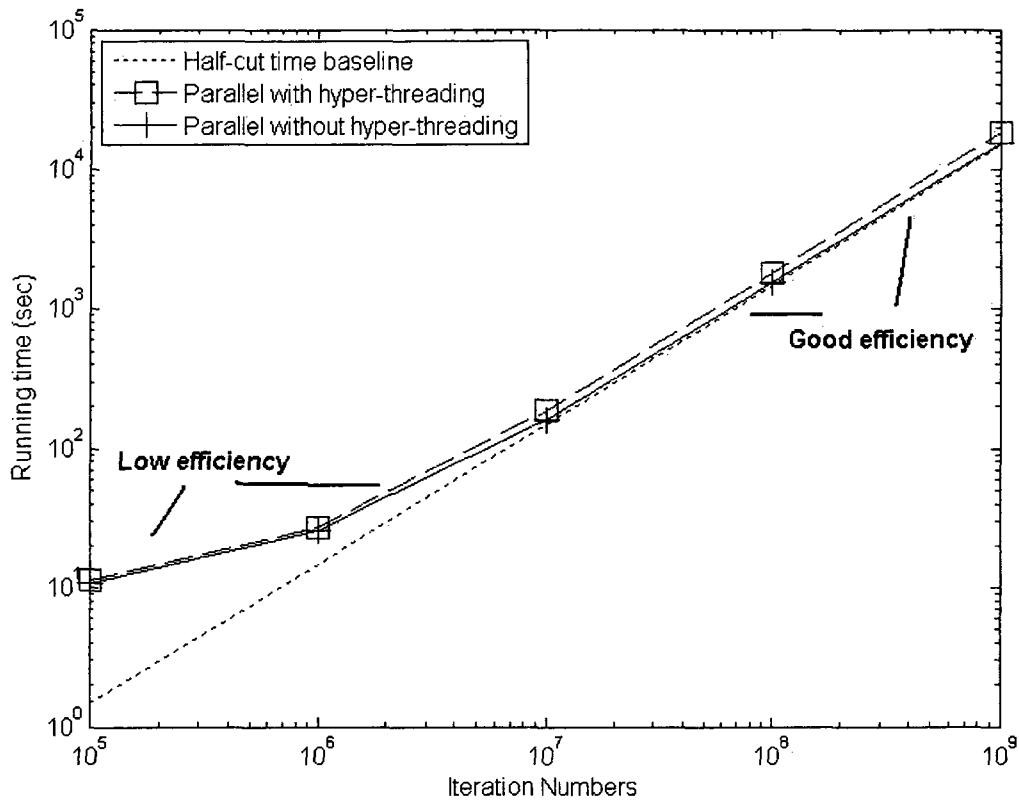


Fig. 8.3 Running time of pMatlab with and without hyper threading vs. 50% running time baseline of the serial simulation

8.3. Summary

The computationally intensive simulation implementations of optimized V-BLAST for various modulations in this thesis show parallel Matlab® computation is meaningful. At the beginning of this chapter, we defined three typical types of parallelism. After an exhaustive investigation into toolkits (detailed in the Appendix.B), we picked pMatlab and PCT™ to facilitate our parallel MC simulation applications. The simulation results substantiated that parallel Matlab® is instrumental in speeding up demanding MC simulation implementations without penalizing estimation accuracy. Nevertheless, implementing the abovementioned toolkits

directly left something to be desired; i.e., the running time isn't halved on a dual processor machine. Hence, we ran elaborate studies on pMatlab parallelism efficiency. Finally, executing disassembling parallelism revealed that the performance gap was due to the toolkit's initializations and the hyper threading operations introduced by the operating system (e.g., Windows XP in our cases.)

Chapter 9. Conclusion

9.1. Summary of the Thesis

V-BLAST algorithms achieve a balance between signal processing complexity and achievable MIMO capacity. Of the different V-BLAST algorithms (Chapter 3), ZF-VBLAST without optimum ordering is a promising option for practical MIMO implementation because of its simplicity. However, error propagation between detection steps leads to an effort to find a low-complexity alternative to the optimal ordering in Foschini's conventional V-BLAST. Although the computational efficiency can be improved somewhat by using a few alternatives, such as partial ML detection, first step ordering and QR decompositions of $\mathbf{H}$, the non-uniform power allocation technique, which achieves similar gain as the optimum ordering does, is chosen for our analysis because of its ready architecture and intuitive strategy. In non-uniform power allocation algorithm, more power is allocated to the streams that correspond to lower decoding steps because the errors from these steps dominate the total error rate.

Kostina has accomplished a frame work to derive the optimum power allocation solutions based on various criteria in [32], but the study was limited to BPSK. This thesis extends the frame work to higher level modulations.

Consequently, the target average BLER expression for optimization was derived as follows. First, we obtained the unified instantaneous SER expression over the single fading channel for different modulation formats by introducing two constellation-dependent parameters, normalized minimum Euclidean distance squared and the average number of the nearest constellation neighbors. Second, we extended this expression to MRC combining over multi-path fading channels. The corresponding closed-form result for BPSK from [50] eventually evolved into a unified SER upper bound for various constellations (Chapter 4). To the best of our knowledge,

this is the first time that asymptotic SER approximation over diversity combining has been quantified by the constellation-based parameters. As [43] attributes the asymptotic average BLER in an unordered ZF-VBLAST to the summation of each step's average SER, the target unified BLER performance function for the power optimization can then be expressed as the summation of the unified SER upper bound for a variety of constellations.

After we got the unified closed form average BLER expression ready, we derived analytical optimization solutions based on this expression (Chapter 5). Besides the first step still dominating in terms of the error rate for different modulations, the optimum solutions for various constellations asymptotically converge to certain constants determined by the number of transceiver antenna branches, even though this convergence slows down as the constellation level increases. Significantly, for power allocations close to $\gamma_0 = 0$ (low SNR), the solutions are only determined by the number of transmission branches. Further, our MC simulations and analysis both indicated that BLER performance improvements converge to $10\log(m)\text{dB}$ asymptotically no matter what constellation is involved. A higher constellation suffers a slower performance gain increase in the meantime.

We also rigorously defined and studied robustness in terms of power allocation variations and the performance gain of the optimization. Although investigations demonstrate the algorithm is sensitive globally when the power deviates from the optimum setting too much and lower constellations suffer when slopes become steeper, the local insensitivity around the optimum settings still holds. Consequently, it can be found that the asymptotic SNR gain is determined by the minimum Euclidean distance, but the gain for small SNR only depends on the diversity order $(n - m + i)$.

In addition, a toolkit useful for our MC simulation implementations, parallel Matlab® (pMatlab), was discussed (Chapter 8). First, three typical parallelism schemes were considered. Second, pMatlab and PCT™ were chosen to facilitate the MC simulation applications in this thesis after exhaustive candidate screening. Third, we studied the run-time performance gap between the actual measures and the theoretical time cut and cautioned users regarding applying parallel Matlab®.

9.2. Future Research

Since this paper inherits the major part of the framework from [32] but just generalizes the work based on the average BLER-based optimization to various constellations, a further avenue of research could be to extend the rest of Kostina's work to various modulation cases, such as average TBER and instantaneous error rate-based optimizations. As the validity of the SER generalization at small and moderate SNR has been demonstrated in this work, the error propagations in a TBER-based optimization can also be investigated with the same generalization.

In addition, because of the heuristic minimum Euclidean distance-parameterized SER quantification, it is obvious that not only the performance measures over Rayleigh channels but also those over other slow-fading channels, e.g., Ricean and log-normal fading, can be quantified into closed-form expressions for power optimization analysis and simulation. Hence, any analysis of the power optimized VBLAST performance over Ricean/log-normal channels can be generalized for different modulations with AUB approximation.

Another extension is the generation from ZF-VBLAST to MMSE-VBLAST. The unified approach in this work can be applied to different V-BLAST architectures as long as the

analytical performance in each layer peeling step can be expressed in a closed form. The major obstacle is the performance evaluation of the output after the MMSE nulling.

For the parallel Matlab® implementation, the efforts could be extended to “cluster parallelisms,” where jobs are shared among computers through an intranet. A few toolkits, such as Mathwork’s PCT, offer this feature and many more interesting observations and studies on the corresponding computation efficiency are expected in the near future.

Reference

- [1] “A standardized set of MIMO radio propagation channels”, Lucent, Nokia, Siemens, Ericsson, Jeju, Korea, 3GPP TSG-RAN WG1 23, N. 1., PP.19–23, 2001
- [2] S.M. Alamouti, “A simple transmit diversity technique for wireless communications”, *IEEE Select. Areas Commun.*, V. 16, N. 8, PP.1451–1458, 1998.
- [3] S. Baro, G. Bauch, A. Hansmann, “Improved codes for space–time trellis–coded modulation”, *IEEE Commun. Letters*, V. 4, N.1, PP. 20–22, 2002.
- [4] J. Barry, E. A. Lee and D. Messerscgnutt, Digital Comm. 3rd edition, Kluwer Academic Publishers, 2004.
- [5] G. Bauch, “Concatenation of space–time block codes and turbo–TCM”, IEEE International Conference on Communications, V. 2, PP.1202–1206, 1999.
- [6] J. Benesty, Y. Huang, J. Chen, “A fast recursive algorithm for optimum sequential signal detection in a BLAST system”, *IEEE Trans. Signal Processing*, V. 51, N.7, PP.1722–1730, 2003.
- [7] H. Bolcskei, M. Borgmann, A.J. Paulraj, “Impact of the propagation environment on the performance of space–frequency coded MIMO–OFDM”, *IEEE Select. Areas Commun.*, V. 21, N. 3, PP.427 – 439, 2003.
- [8] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004
- [9] N. Chiurtu, B. Rimoldi, and E. Telatar, “Dense multiple antenna systems”, IEEE Information Theory Workshop, PP.108–109, 2001.
- [10] D. Chizhik, G. J. Foschini, M. J. Gans, and R. A. Valenzuela, “Keyholes, correlations and capacities of multielement transmit and receive antennas”, *IEEE Trans. Wireless Commun.*, V. 1, N. 2, PP.361–368, 2002.

- [11] Won-Joon Choi, R. Negi, J.M. Cioffi, "Combined ML and DFE decoding for the V-BLAST system", *IEEE International Conference on Communications*, V.3 PP.1243–1248, 2000.
- [12] Won-Joon Choi, Kok-Wui Cheong, J.M. Cioffi, "Iterative soft interference cancellation for multiple antenna systems", *IEEE Wireless Communications and Networking Conference*, V.1, PP.304–309, 2000.
- [13] R. Choy and A. Edelman, "Parallel MATLAB: Doing It Right", *Proceeding of the IEEE*, V.93, N.2, 2005.
- [14] M. Coupechoux and V. Braun, "Space time coding for the EDGE mobile radio system", *IEEE International Conference on Personal Wireless Communications*, PP 28–32, 2000.
- [15] J. Craig, "A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations", *IEEE Military Communications Conference*, V.2, PP.571–575, 1991.
- [16] R.T. Derryberry, S.D. Gray, D.M. Ionescu, G. Mandyam, and B. Raghothaman, "Transmit diversity in 3G CDMA systems", *IEEE Communications Magazine*, V.40, N.4, PP. 68–75, 2002.
- [17] P. F. Driessen and G. J. Foschini, "On the capacity formula for multiple-input multiple-output wireless channels: A geometric interpretation", *IEEE International Conference on Personal Communications*, V.3, PP.1603–1607, 1999.
- [18] F. R. Farrokhi, G. J. Foschini, A. Lozano, and R. A. Valenzuela, "Link optimal space-time processing with multiple transmit and receive antennas", *IEEE Commun. Letter*, V.5, PP.85–87, 2001.
- [19] G.J. Foschini, "Layered space-time architecture for wireless communication in fading environment when using multiple antenna", *Bell Lab Tech. Journal*, V.1, N.2, PP.41–59, 1996.

- [20] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas", *Wireless Personal. Commun.*, V.6, PP.311–335, 1998.
- [21] G. J. Foschini, "Analysis and performance of some basic space–time architectures", *IEEE Select. Areas Commun.*, V.21, N.3, PP.281–320, 2003.
- [22] D. Gesbert, H. Bolcskei, D. Gore, and A. Paulraj, "Outdoor MIMO wireless channels: Models and performance prediction", *IEEE Trans. Commun.*, V.50, N.12, PP.1926 – 1934, 2002.
- [23] D. Gesbert, M. Shafi, D.S. Shiu and et.al. "From theory to practice: an overview of MIMO space–time coded wireless systems", *IEEE Select. Areas Commun.*, V.21, N.3, 2005.
- [24] Y. Gong, K.B. Letaief, "Concatenated space–time block coding with trellis coded modulation in fading channels", *IEEE Trans. Wireless Commun.*, V.1, N.4, PP.580–590, 2002.
- [25] A. Gorokhov, D. Gore, A. Paulraj, "Receive antenna selection for MIMO flat–fading channels: theory and algorithms", *IEEE Trans. Information Theory*, V.49, N.10, PP:2687 – 2696, 2003.
- [26] J.H. Jang, H.C. Won and G.H. Im, "Cyclic prefixed single carrier transmission with SFBC over mobile wireless channels", *IEEE Signal Processing Letter*, V.13, N.5, 2006.
- [27] M. Jeruchinm, P. Balaban and K. S. Shanmugan, *Simulation of Communication System*, 2nd edition, Kluwer Academic, 2000.
- [28] R. Kalbasi, D. D. Falconer, and A. H. Banihashemi, "Optimum power allocation for a V–BLAST system with two antennas at the transmitter", *IEEE Commun. Letter*, V.9, N.9, PP.826–828, 2005.
- [29] M. Kang and M.S. Alouini, "Capacity of correlated MIMO Rayleigh channels", *IEEE Trans. Wireless Commun.*, V.5, N.1, PP.143–155, 2006.

- [30] M. Kang and M.S. Alouini, "Capacity of MIMO Rician channels", *IEEE Trans. Wireless Commun.*, V.5, N.1, PP.112–122, 2006.
- [31] J. Kim, S.L. Ariyavisitakul and N. Seshadri, "STBC/SFBC for 4 Transmit Antennas with 1-bit Feedback", IEEE International Conference on Communications, PP.3943 – 3947, 2008.
- [32] V. Kostina and S. Loyka, "On Optimum Power Allocation for the V-BLAST", *IEEE Trans. on Commun.*, V. 56, N. 6, PP. 999–1012, 2008,
- [33] B. Hassibi, "A fast square-root implementation for BLAST", Thirty-Fourth Asilomar Conf. Signals, Syst. Comput., PP.1255–1259, 2000.
- [34] E. Larsson, P. Stoica, Space-Time Block Coding for Wireless Communications, Cambridge University Press, 2003.
- [35] Z. Latinovic and Y. Bar-Ness, "SFBC MIMO-OFDM peak-to-average power ratio reduction by polyphase interleaving and inversion", IEEE Commun. Letter, V.4, N.10, PP.266–268, 2006.
- [36] G. Lebrun, M. Faulkner, M. Shafi and P.J. Smith, "MIMO Ricean channel capacity: an asymptotic analysis", IEEE Trans. Wireless Commun., V.5, N.6, PP.1343 – 1350, 2006.
- [37] G. Levin and S. Loyka, "On the outage capacity distribution of correlated keyhole MIMO channels", *IEEE Trans. Information Theory*, V.54, N.7, 2008.
- [38] L.H. Li, X.F. Tao, P. Zhang and H. Haas, "A practical space-frequency block coded OFDM scheme for fast fading broadband channels", IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, V.1, PP.212 – 216, 2002.
- [39] X.B. Liang and X.G. Xia, "On the nonexistence of rate-one generalized complex orthogonal designs", *IEEE Trans. Information Theory*, V.49, N.11, PP.2984 – 2988, 2003.

- [40] E. Lindskog and A. Paulraj. "A transmit diversity scheme for channels with intersymbol interference", *IEEE International Conference on Communications*, PP.307–311, 2000.
- [41] Z. Liu, G.B. Giannakis, S. Zhou and B. Muquet, "Space-time coding for broadband wireless communications", *Wireless Communications and Mobile Computing*, V.1, N.1 PP.35–53, 2001.
- [42] S. Loyka and F. Gagnon, "Performance analysis of the V-BLAST algorithm: An analytical approach", *IEEE Trans. Wireless Commun.*, V.2, N.4, PP.1326–1337, 2004.
- [43] S. Loyka and F. Gagnon, "V-BLAST without optimal ordering: analytical performance evaluation for Rayleigh fading channels", *IEEE Trans. Commun.*, V.54, N.6, PP.1109–1120, 2006.
- [44] J. Lu, T.T. Tjhung, and C.C. Chai, "Error probability performance of L-branch diversity reception of MQAM in Rayleigh fading", *IEEE Trans. Commun.*, V.46, N.2, 1998.
- [45] M.R. McKay, I.B. Collings, "Improved general lower bound for spatially-correlated Rician MIMO capacity", *IEEE Commun. Letter*, V.10, N.3, PP.162–164, 2006.
- [46] S. H. Nam et al., "Transmit power allocation for a modified V-BLAST system", *IEEE Trans. Commun.*, V.52, N.7, PP.1074–1079, 2004.
- [47] C. Oestges and V. Erceg, A.J. Paulraj, "A physical scattering model for MIMO macrocellular broadband wireless channels", *IEEE Select. Areas Commun.*, V.21, N.5, PP.721 – 729, 2003.
- [48] A. Paulraj, R. Nabar and D. Gore, *Introduction to space-time wireless communications*, Cambridge University Press, 2003.

- [49] N. Prasad and M. K. Varanasi, "Analysis of decision feedback detection for MIMO Rayleigh-fading channels and the optimization of power and rate allocations", *IEEE Trans. Information Theory*, V.50, N.6, PP. 1009 – 1025, 2004.
- [50] J. G. Proakis, *Digital Communications*. Boston: McGraw-Hill, 2001.
- [51] P.B Rapajie and D. Popeseu, "Information capacity of a random signature multiple –input multiple–output channel", *IEEE Trans. Commun.*, V.48, N.8, PP.1245–1248, 2000.
- [52] T.S. Rappaport. *Wireless Communications, Principles and Practices*, 2nd Edition, Pearson Education, 2002.
- [53] C. Shen, H. Zhuang, I. Dai and S. Zhou, "Detection algorithm improving V-BLAST performance over error propagation", *IEEE Electronics Letters*, V.39, N.13, PP.1007–1008, 2003.
- [54] Z.K. Shen, R.H. Chen, J.G. Andrews, R.W. Heath and B.L. Evans, "Sum Capacity of Multiuser MIMO Broadcast Channels with Block Diagonalization", *IEEE Trans. Commun.*, V.6, N.6 , PP.2040 – 2045. 2007.
- [55] D. Shiu, *Wireless Communication Using Dual Antenna Arrays*, International Series in Engineering and Computer Science. Norwell, 1999.
- [56] D. Shiu, G. J. Foschini, M. J. Gans, and J. M. Kahn, "Fading correlation and its effect on the capacity of multielement antenna systems", *IEEE Trans. Commun.*, V.48, N.3, PP.502–513, 2000.
- [57] M. K. Simon, M. S. Alouini, *Digital Communication over Fading Channels*, Wily–Interscience Publication, 2000.

- [58] S. Siwamogsatam and M. Fitz, “Improved high-rate space-time codes via orthogonality and set partitioning”, IEEE Wireless Communications and Networking Conference, V: 1, PP.264– 270, 2002.
- [59] Xin Ng Soon and L. Hanzo, “On the MIMO channel capacity of multidimensional signal sets”, *IEEE Trans. Vehicular Technology*, V.55, N. 2, PP.528– 536, 2006.
- [60] R. Stridh, Kai Yu, B. Ottersten and P. Karlsson, “MIMO channel capacity and modeling issues on a measured indoor radio channel at 5.8 GHz”, *IEEE Trans. Wireless Commun.*, V.4, N.3, PP.895 – 903, 2005.
- [61] G. L. Stuber, Principles of Mobile Communication, 2nd Edition, Kluwer Academic Publishers, 2001.
- [62] V. Tarokh, N. Seshadri, and A. R. Calderbank, “Space-time codes for high data rate wireless communication: Performance analysis and code construction”, *IEEE Trans. Information Theory*, V.44, N.2, PP.744–765, 1998.
- [63] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, “Space-time block codes from orthogonal designs”, *IEEE Trans. Information Theory*, V. 45, N.5, PP.1456–1467, 1999.
- [64] I. E. Telatar, “Capacity of multiantenna Gaussian channels”, *Eur. Trans. Commun.*, V.10, N.6, PP.585–595, 1999.
- [65] W. H. Tranter, K. S. Shanmugan and et., al. Principles of Communication Systems Simulation with Wireless Applications, Prentice Hall, 2004.
- [66] D. Tse and P. Viswanth, Fundamentals of Wireless Communication, Cambridge University Press, 2005.

- [67] P. Viswanath, D. N. C. Tse, and V. Anantharam, "Asymptotically optimal water-filling in vector multiple-access channels", *IEEE Trans. Information Theory*, V.47, N.1, PP.241–267, 2001.
- [68] Zh. Wang and G. Giannakis, "A simple and general parameterization quantifying performance in fading channels", *IEEE Trans. Commun.*, V.51, N.8, 2003.
- [69] N. Wang and S. D. Blostein, "Comparison of CP-based single-carrier and OFDM with power allocation", *IEEE Trans. Commun.*, V.53, N.3, PP.391–394, 2005.
- [70] N. Wang and S.D. Blostein, "Approximate Minimum BER Power Allocation for MIMO Spatial Multiplexing System", *IEEE Trans. Commun.*, V. 55, N. 1, PP.180–187, 2007.
- [71] J.H. Winter, "On the capacity of radio communication systems with diversity in a Rayleigh fading environment", *IEEE Select. Areas Commun.* V. SAC–5, N.5, PP.871–878, 1987.
- [72] P.W. Wolniansky, G. J. Foschini and et., al. "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel", *Proc. Int. Symposium on Advanced Radio Tech.*, 1998.
- [73] Y. Xie and C.N. Georghiades, "Some results on the sum-rate capacity of MIMO fading broadcast channels", *IEEE Trans. Wireless Commun.*, V.5, N.2, PP.377 – 383, 2006.
- [74] Z. Yan, K.M. Wong and Z.Q. Luo, "Optimal diagonal precoder for multiantenna communication systems", *IEEE Trans. Signal Processing*, V.53, N.6, PP.2089–2100, 2005.
- [75] H.W. Yang, "A road to future broadband wireless access: MIMO-OFDM-Based air interface", *IEEE Communications Magazine*, V.43, N.1, PP.53–60, 2005.
- [76] W. Zha and SD Blostein, "Modified decorrelating decision-feedback detection of BLAST space-time system", *IEEE International Conference on Communications*, PP.335–339, 2002.

[77] “Channel models for fixed wireless Applications”, IEEE, Tech. Rep., IEEE 802.16 Work Group, 2001.

Appendix A. Mathematical Derivations

Appendix A1. Weighting Factor in MMSE-VBLAST

1. Max SNR VBLAST

Consider the output SNR

$$\gamma_i = \frac{P_s}{P_\xi} = \frac{\mathbf{w}_i^+ \mathbf{h}_i' \mathbf{h}_i^{*+} \mathbf{w}_i}{\mathbf{w}_i^+ (\sigma_0^2 \mathbf{I} + \mathbf{H}_{(i)} \mathbf{H}_{(i)}^+) \mathbf{w}_i} \quad (\text{A1.1})$$

The solution requires

$$\max_{\mathbf{w}_i} (\gamma_i)$$

Denote

$$\mathbf{R}_\xi = \sigma_0^2 \mathbf{I} + \mathbf{H}_{(i)} \mathbf{H}_{(i)}^+$$

Following the Cauchy -Schwarz inequality

$$\gamma_i = \frac{\mathbf{w}_i^+ \mathbf{h}_i' \mathbf{h}_i^{*+} \mathbf{w}_i}{\mathbf{w}_i^+ (\sigma_0^2 \mathbf{I} + \mathbf{H}_{(i)} \mathbf{H}_{(i)}^+) \mathbf{w}_i} = \frac{|\mathbf{w}_i^+ \mathbf{h}_i'|^2}{|\mathbf{w}_i^+ \mathbf{R}_\xi^{1/2}|^2} = \frac{|\mathbf{w}_i^+ \mathbf{R}_\xi^{1/2} \mathbf{R}_\xi^{-1/2} \mathbf{h}_i'|^2}{|\mathbf{w}_i^+ \mathbf{R}_\xi^{1/2}|^2} \leq |\mathbf{R}_\xi^{-1/2} \mathbf{h}_i'|^2$$

To make above equation stand, the following condition should be satisfied

$$\mathbf{w}_i^+ \mathbf{R}_\xi^{1/2} = (k \mathbf{R}_\xi^{-1/2} \mathbf{h}_i')^+ \rightarrow \mathbf{w}_i = k \mathbf{R}_\xi^{-1} \mathbf{h}_i' \quad (\text{A1.2})$$

2. MMSE VBLAST

To minimize the mean square error

$$\varepsilon_i^2 = \left\langle |s_i' - \hat{s}_i'|^2 \right\rangle = \left\langle |s_i' - \mathbf{w}_i^+ \mathbf{r}|^2 \right\rangle$$

The derivation of above error is

$$\begin{aligned}\frac{d\epsilon_i^2}{d\mathbf{w}_i} &= \frac{d\langle |s'_i - \mathbf{w}_i^+ \mathbf{r}'|^2 \rangle}{d\mathbf{w}_i} = \frac{d\langle (s'_i - \mathbf{w}_i^+ \mathbf{r}') \cdot (s'^*_i - \mathbf{r}'^+ \mathbf{w}_i) \rangle}{d\mathbf{w}_i} \\ &= \frac{d\langle s'_i s'^*_i - \mathbf{w}_i^+ \mathbf{r}' s'^*_i - s'_i \mathbf{r}'^+ \mathbf{w}_i + \mathbf{w}_i^+ \mathbf{r}' \mathbf{r}'^+ \mathbf{w}_i \rangle}{d\mathbf{w}_i}\end{aligned}$$

Because $\langle s'_i s'^*_i \rangle = 1$, above derivation goes to

$$\frac{d\epsilon_i^2}{d\mathbf{w}_i} = \frac{d(1 - \langle \mathbf{w}_i^+ \mathbf{r}' s'^*_i \rangle - \langle s'_i \mathbf{r}'^+ \mathbf{w}_i \rangle + \langle \mathbf{w}_i^+ \mathbf{r}' \mathbf{r}'^+ \mathbf{w}_i \rangle)}{d\mathbf{w}_i} = -\langle s'_i \mathbf{r}'^+ \rangle + \langle \mathbf{w}_i^+ \mathbf{r}' \mathbf{r}'^+ \rangle = 0 \quad (\text{A1.3})$$

Thus

$$\mathbf{w}_i^+ = \langle s'_i \mathbf{r}'^+ \rangle \langle \mathbf{r}' \mathbf{r}'^+ \rangle^{-1} \rightarrow \mathbf{w}_i = \langle \mathbf{r}' \mathbf{r}'^+ \rangle^{-1} \langle \mathbf{r}' s'^*_i \rangle$$

Assume s'_i and s'_j are uncorrelated when $i \neq j$, from the definition of $\mathbf{r}'$

$$\mathbf{r}' = \mathbf{H}_{(i-1)} \mathbf{S}_{(i-1)} + \xi = \begin{bmatrix} h'_{1i} s'_i + h'_{1(i+1)} s'_{i+1} + \dots + h'_{1m} s'_m + \xi_1 \\ h'_{2i} s'_i + h'_{2(i+1)} s'_{i+1} + \dots + h'_{2m} s'_m + \xi_2 \\ \vdots \\ h'_{ni} s'_i + h'_{n(i+1)} s'_{i+1} + \dots + h'_{nm} s'_m + \xi_n \end{bmatrix}$$

It is obvious that $\langle \mathbf{r}' s'^*_i \rangle = \mathbf{h}_i$

3. PDF of the Output SNR in i th Step

Since $|\gamma_i|^2 \sim \chi^2_{2 \times (n-m+i)}$, the normalized version γ_i / γ_0 has the cumulative distribution function

(CDF)

$$\begin{aligned}\Pr\{\gamma_i / \gamma_0 < x\} &= 1 - e^{-x} \sum_{k=0}^{n-m+i-1} x^k / k! \\ &= 1 - \left(1 - x + \frac{(-x)^2}{2!} + \frac{(-x)^3}{3!} + \dots \right) \times \left[\frac{x^0}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^{n-m+i-1}}{(n-m+i-1)!} \right]\end{aligned} \quad (\text{A1.4})$$

The items in the polynomial production cancel with each other and only the last term

$x^{n-m+i-1} / (n-m+i-1)!$ left. Thus, the CDF is

$$\Pr\{\gamma_i / \gamma_0 < x\} = x^{n-m+i-1} / (n-m+i-1)! \quad (\text{A1.5})$$

For the PDF of the normalized γ_i / γ_0 , the derivation is

$$\begin{aligned} f_i(x) &= F'_i(x) = \left(1 - e^{-x} \sum_{k=0}^{n-m+i-1} x^k / k! \right)', \\ &= e^{-x} \times \left[\frac{x^0}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^{n-m+i-1}}{(n-m+i-1)!} \right] - e^{-x} \left(0 + 1 + \frac{2x}{2} + \dots + \frac{x^{n-m+i-2}}{(n-m+i-2)!} \right) \\ &= e^{-x} \frac{x^{n-m+i-1}}{(n-m+i-1)!} \end{aligned}$$

and its asymptotic approximation is

$$f_i(x) \approx \frac{x^{n-m+i-1}}{(n-m+i-1)!} \quad (\text{A1.6})$$

Appendix A2. Simplified Union Bound of Instantaneous SER

The minimum Euclidean distance in MPSK and MQAM constellations are $d_{\min}$

1. MPSK

From FigA1, assume the average energy of base band signal is E_{av} , then

$$d_{\min} = 2d = 2\sqrt{E_{av}} \sin(\pi / M)$$

So the symbol error probability (SEP) of two adjacent constellations with the distance $d_{\min}$ is

$$\mathcal{Q}\left(\sqrt{\frac{d_{\min}^2}{2N_0}}\right) = \mathcal{Q}\left(\sqrt{\frac{4E_{av} \sin^2(\pi / M)}{2N_0}}\right) \quad (\text{A2.1})$$

Assume the average SNR per symbol is given by

$$\gamma_s = \frac{E_{av}}{N_0} \quad (\text{A2.2})$$

Substituting (A2.2) into (A2.1), one can get the SEP for each nearest neighbor pair

$$\mathcal{Q}\left(\sqrt{2\gamma_s} \sin(\pi/M)\right) \quad (\text{A2.3})$$

Furthermore, if we normalize the constellation properly to make $E_{av} = 1$, we will have a

normalized minimum Euclidean distance $d_{\min}^* = 2 \sin(\pi/M)$ and SEP will be $\mathcal{Q}\left(\sqrt{2\gamma_s} d_{\min}^*/2\right)$.

2. MQAM

The E_{av} in optimum 8-QAM and 16-QAM can be derived from the constellations in FigA2 as

$$E_{av,8\text{QAM}} = d^2 \frac{1}{8} \left[(\sqrt{5})^2 \times 4 + 1 \times 2 + 3^2 \times 4 \right] = 5d^2 = \frac{5d_{\min}^2}{4}$$

$$E_{av,16\text{QAM}} = d^2 \frac{1}{16} [2 \times 4 + 10 \times 8 + 18 \times 4] = 10d^2 = \frac{5d_{\min}^2}{2}$$

Correspondingly, the SNR for them are

$$\gamma_{s,8\text{QAM}} = \frac{E_{av,8\text{QAM}}}{N_0} = \frac{5d^2}{N_0} = \frac{5d_{\min}^2}{4N_0}$$

$$\gamma_{s,16\text{QAM}} = \frac{E_{av,16\text{QAM}}}{N_0} = \frac{10d^2}{N_0} = \frac{5d_{\min}^2}{2N_0}$$

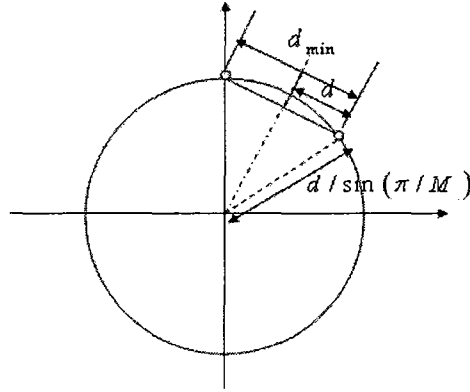


Fig. A1 Constellation of MPSK

SEP for the pair of nearest neighbors then turns to

$$Q\left(\sqrt{\frac{2\gamma_s}{5}}\right) \quad (A2.4)$$

$$Q\left(\sqrt{\frac{\gamma_s}{5}}\right) \quad (A2.5)$$

And the normalized $d_{\min 8QAM}^* = 2/\sqrt{5}$ and $d_{\min 16QAM}^* = \sqrt{2}/5$. The SEP equals to $Q\left(\sqrt{2\gamma_s}d_{\min}^*/2\right)$ as well.

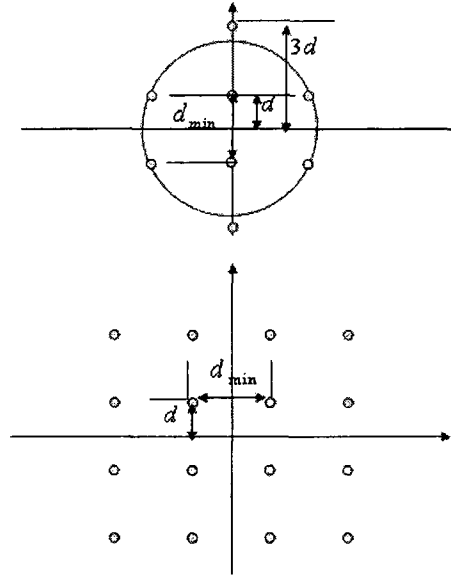


Fig. A2 Constellations of 8-QAM and 16-QAM

Appendix A3. Asymptotic Approximation of SER over Diversity

Substituting conditional SER bound into Wang's asymptotic model for diversity system performance approximation (MRC over diversity i.i.d. Rayleigh fading).

1. Wang's model

For any diversity fading system satisfying following conditions,

- The conditional SNR at receiver can be represented by $\gamma = x \gamma_0$, γ_0 is the average SNR per channel and x is the channel dependent variable, which is $x = \gamma_s / \gamma_0$, γ_s is the instantaneous SNR
- Instantaneous SER may given by $Q(\sqrt{k \cdot x \gamma_0})$ and k is modulation-depended constant.
- The PDF of x can be approximated by a series as $p(x) = ax^t + o(x^{t+\epsilon})$ and $o(x^{t+\epsilon})$ is significant small items which can be neglected.

The asymptotic approximation of average SER is

$$\bar{P}_e \approx (G \gamma_0)^{-D} \quad (\text{A3.1})$$

where the “coding gain” G and the “diversity gain” D are given by

$$\begin{cases} G = k \left(\frac{2' a \Gamma(t+1.5)}{\sqrt{\pi}(t+1)} \right)^{-\frac{1}{(t+1)}} \\ D = t + 1 \end{cases} \quad (\text{A3.2})$$

2. MRC performance

For the Rayleigh fading, because the PDF of x equals to $f(\gamma_s / \gamma_0) = \gamma_0 f_{\gamma_s}(\gamma_s) = \gamma_0 (e^{-\gamma_s / \gamma_0} / \gamma_0) \approx 1 - \gamma_s / \gamma_0$, $a = 1$ and $t = 0$. Then one may have

$$\begin{cases} G = k \left(\frac{\Gamma(1.5)}{\sqrt{\pi}} \right)^{-1} \\ D = 1 \end{cases}$$

and

$$\bar{P}_e \approx (G \gamma_0)^{-D} = \frac{\Gamma(1.5)}{k \sqrt{\pi}} \frac{1}{\gamma_0}$$

Apply this model to MRC performance estimation. When the PDF of $x = \gamma_s / \gamma_0$ after MRC combining is given by

$$f_{\gamma_s/\gamma_0}(\gamma_s/\gamma_0) = \gamma_0 f_{\gamma_s}(\gamma_s) = \gamma_0 \frac{1}{(L-1)! \gamma_0^L} \gamma_s^{L-1} e^{-\gamma_s/\gamma_0} \approx \frac{(\gamma_s/\gamma_0)^{L-1}}{(L-1)!}$$

one can get $a=1/(L-1)!$, $t=L-1$ and

$$\begin{cases} G_{\text{MRC}} = k \left(\frac{2^{L-1} \Gamma(L+0.5)}{\sqrt{\pi} \cdot L!} \right)^{\frac{1}{L}} \\ D_{\text{MRC}} = L \end{cases} \quad (\text{A3.3})$$

The final average SER goes to

$$\overline{P_e} \approx (G_{\text{MRC}} \gamma_0)^{-D_{\text{MRC}}} = \frac{1}{2\sqrt{\pi}} \left(\frac{2}{k\gamma_0} \right)^L \frac{\Gamma(L+0.5)}{\Gamma(L+1)} \quad (\text{A3.4})$$

Modifying $Q(\sqrt{k \cdot x\gamma_0})$ into $N_c Q(\sqrt{D_s \cdot 2x\gamma_0})$ with two constellation-dependent parameters D_s and N_c , this model turns to

$$\begin{cases} G_{\text{MRC}}^* = 2D_s \left(N_c \frac{2^{L-1} \Gamma(L+0.5)}{\sqrt{\pi} \cdot L!} \right)^{\frac{1}{L}} \\ D_{\text{MRC}}^* = L \end{cases} \quad (\text{A3.5})$$

and

$$\overline{P_e} \approx (G_{\text{MRC}}^* \gamma_0)^{-D_{\text{MRC}}^*} = \frac{N_c}{2\sqrt{\pi}} \left(\frac{1}{D_s \gamma_0} \right)^L \frac{\Gamma(L+0.5)}{\Gamma(L+1)} \quad (\text{A3.6})$$

Appendix A4. Asymptotic Approximation of V-BLAST BLER

The BLER of a non-uniform power allocation VBLAST is

$$\begin{aligned} \overline{P_B}(\alpha) &\approx \sum_{i=1}^m \overline{P_{ei}} \\ &\approx \sum_{i=1}^m N_c \left[\frac{1}{2} (1 - \mu_i') \right]^{n-m+i} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \left[\frac{1}{2} (1 + \mu_i') \right]^k \end{aligned} \quad (\text{A4.1})$$

where

$$\mu_i' = \sqrt{\frac{\alpha_i D_s \gamma_0}{1 + \alpha_i D_s \gamma_0}}$$

In each branch i , the average SNR of transmission signal turns from γ_0 of uniform power allocation to $\alpha_i \gamma_0$ of non-uniform allocation.

Because there are $\mu_i' \approx 1$ and $(1 + \mu_i')/2 \approx 1$ for the asymptotic cases, one may also have

$$\frac{1}{2}(1 - \mu_i') \approx \frac{1}{2} \left(1 - \sqrt{\frac{\alpha_i D_s \gamma_0}{1 + \alpha_i D_s \gamma_0}} \right) \frac{1}{2} \left(1 + \sqrt{\frac{\alpha_i D_s \gamma_0}{1 + \alpha_i D_s \gamma_0}} \right) = \frac{1}{4} \left(\frac{1}{1 + \alpha_i D_s \gamma_0} \right) \approx \frac{1}{1 + 4\alpha_i D_s \gamma_0}$$

Since the summation of binomial coefficients follows:

$$\sum_{k=0}^{n-m+i-1} C_{n-m+i-1+k}^k = C_{2(n-m+i)-1}^{n-m+i}$$

(A4.1) is finally approximated by

$$\overline{P_B}(\alpha) \approx \sum_{i=1}^m N_c \left[\frac{1}{4\alpha_i D_s \gamma_0} \right]^{n-m+i} C_{2(n-m+i)-1}^{n-m+i} \quad (\text{A4.2})$$

Appendix A5. Optimum Power Allocation Coefficients

Derive the optimum power allocation coefficient α_{opt} when below derivation of Lagrangian equals to zero

$$\frac{\partial L(\alpha)}{\partial \alpha_i} = -N_c \cdot \frac{n-m+i}{(\alpha_i)^{n-m+i+1}} \frac{C_{2(n-m+i)-1}^{n-m+i}}{(4D_s \gamma_0)^{n-m+i}} + \lambda \quad (\text{A5.1})$$

The answer is

$$\begin{aligned} \alpha_{i,\text{opt}} &= \sqrt[n-m+i+1]{\frac{N_c(n-m+i)C_{2(n-m+i)-1}^{n-m+i}}{(4D_s \gamma_0)^{n-m+i} \lambda}} \\ &= \frac{a_i}{\sqrt[n-m+i+1]{(4D_s \gamma_0)^{n-m+i} \lambda}} \end{aligned} \quad (\text{A5.2})$$

where

$$a_i = \sqrt[n-m+i+1]{N_c (n-m+i) C_{2(n-m+i)-1}^{n-m+i}}$$

Substituting $\alpha_{i,\text{opt}}$ into the condition:

$$\sum_{i=1}^m \alpha_i = m$$

one may get

$$\sum_{i=1}^m a_i (4D_s \gamma_0)^{-\frac{n-m+i}{n-m+i+1}} \lambda^{-\frac{1}{n-m+i+1}} = m$$

If we define

$$\begin{cases} c_i = \frac{(n+1)!}{(n-m+1)!(n-m+i+1)} \\ x = \lambda^{-(n-m+1)!/(n+1)!} \end{cases}$$

above summation becomes

$$\sum_{i=1}^m a_i (4D_s \gamma_0)^{-\frac{n-m+i}{n-m+i+1}} x^{c_i} = m$$

The 1st term of above series can be regarded as a reasonable guess of the solutions of x .

Following Newton Raphson method, one has the 1st approximation given by

$$a_1 (4D_s \gamma_0)^{-\frac{n-m+1}{n-m+2}} x_0^{c_1} = m$$

$$x_0 = \sqrt[n-m+2]{m (4D_s \gamma_0)^{\frac{n-m+1}{n-m+2}} / a_1}$$

Using $x = x_0(1+\delta)$ as the 2nd order approximation and keeping the first two terms of the series

$$\sum_{i=1}^m a_i (4D_s \gamma_0)^{-\frac{n-m+i}{n-m+i+1}} x_0^{c_i} (1+\delta)^{c_i} = m$$

and substituting x_0 into this truncated series, one may have

$$\begin{aligned} a_i x_0^{c_i} &= a_i m^{c_i/c_1} a_1^{-c_i/c_1} (4D_s \gamma_0)^{(n-m+1)c_i/(n-m+2)c_1} \\ &= a_i m^{(n-m+2)/(n-m+i+1)} a_1^{-(n-m+2)/(n-m+i+1)} (4D_s \gamma_0)^{(n-m+1)/(n-m+i+1)} \\ &= b_i (4D_s \gamma_0)^{(n-m+1)/(n-m+i+1)} \end{aligned}$$

by defining

$$\begin{aligned} b_i &= a_i m^{(n-m+2)/(n-m+i+1)} a_1^{-(n-m+2)/(n-m+i+1)} \\ &= \sqrt[n-m+i+1]{\frac{N_c(n-m+i)C_{2(n-m+i)-1}^{n-m+i} \cdot m^{n-m+2}}{N_c(n-m+1)C_{2(n-m+1)-1}^{n-m+1}}} \end{aligned}$$

Thus, the first two terms of $\sum_{i=1}^m b_i (1+\delta)^{c_i} (4D_s \gamma_0)^{(1-i)/(n-m+i+1)} = m$ can offer us an approximation

$$m(1+\delta)^{c_1} + b_2(1+\delta)^{c_2} (4D_s \gamma_0)^{-1/(n-m+3)} \approx m$$

Because $c_2 < c_1$ and δ is small, one may further approximate above approximation as

$$m(1+c_1\delta) + b_2(4D_s \gamma_0)^{-1/(n-m+3)} \approx m$$

and get the solution

$$\begin{cases} \delta \approx -\frac{b_2}{mc_1} (4D_s \gamma_0)^{-1/(n-m+3)} \\ x \approx \sqrt[n-m+i+1]{m(4D_s \gamma_0)^{\frac{n-m+1}{n-m+2}} / a_1} \left[1 - \frac{b_2}{mc_1} (4D_s \gamma_0)^{-1/(n-m+3)} \right] \end{cases}$$

Finally, the Lagrange multiplier λ is given by

$$\lambda = x^{-\frac{(n+1)!}{(n-m+1)!}} \approx \frac{1}{(4D_s \gamma_0)^{(n-m+1)}} \left(\frac{a_1}{m} \right)^{n-m+2} \left[1 - \frac{b_2}{mc_1^{n-m+3} \sqrt[n-m+3]{4D_s \gamma_0}} \right]^{-(n-m+2)c_1} \quad (\text{A5.3})$$

By substituting λ into (A5.2), optimum power allocation coefficients is given by

$$\tilde{\alpha}_{i,\text{opt}} \approx b_i (4D_s \gamma_0)^{(1-i)/(n-m+i+1)} \left[1 - \frac{b_2}{mc_1^{n-m+3} \sqrt[n-m+3]{4D_s \gamma_0}} \right]^{c_i}$$

Because $b_2 / (mc_1^{n-m+3} \sqrt[n-m+3]{4D_s \gamma_0}) \ll 1$, the further approximation goes to

$$\begin{cases} \tilde{\alpha}_{i,\text{opt}} \approx b_i (4D_s \gamma_0)^{(1-i)/(n-m+i+1)} \\ \alpha_{i,\text{opt}} = m \tilde{\alpha}_{i,\text{opt}} / \sum_{k=1}^m \tilde{\alpha}_{k,\text{opt}} \\ \alpha_{1,\text{opt}} = m - \sum_{i=2}^m \alpha_{i,\text{opt}} \end{cases} \quad (\text{A5.4})$$

where

$$b_i = \frac{(n-m+i)C_{2(n-m+i)-1}^{n-m+i} \cdot m^{n-m+2}}{(n-m+1)C_{2(n-m+1)-1}^{n-m+1}} \quad (A5.5)$$

$$c_i = \frac{(n+1)!}{(n-m+1)!(n-m+i+1)}$$

Appendix A6. Derivative of $\bar{P}_{ei}$ Around $\gamma_0 = 0$

For the truncated MacLaurin's series of

$$\begin{aligned} \bar{P}_{ei} &\approx \frac{1}{2} + \frac{\partial \bar{P}_{ei}}{\partial \sqrt{\alpha_i D_s \gamma_0}} \bigg|_{\sqrt{\alpha_i D_s \gamma_0} = 0} \cdot \sqrt{\alpha_i D_s \gamma_0} \\ &= \frac{1}{2} + d_i \cdot \sqrt{\alpha_i D_s \gamma_0} \end{aligned} \quad (A6.1)$$

the coefficient d_i is given by

$$d_i = \frac{\partial \left\{ N_c \cdot \left[\frac{1}{2} \left(1 - \sqrt{\frac{x^2}{1+x^2}} \right) \right]^{n-m+i} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \left[\frac{1}{2} \left(1 + \sqrt{\frac{x^2}{1+x^2}} \right) \right]^k \right\}}{\partial x} \bigg|_{x=0}$$

where $x = \sqrt{\alpha_i D_s \gamma_0}$. Above derivative further comes to

$$\begin{aligned} d_i &= \left\{ N_c (n-m+i) \left[\frac{1}{2} \left(1 - \sqrt{\frac{x^2}{1+x^2}} \right) \right]^{n-m+i-1} \left(-\frac{1}{2} \right) \frac{1}{2} \left(\frac{x^2}{1+x^2} \right)^{-1/2} \left[\frac{2x}{1+x^2} - \frac{x^3 \cdot 2x}{(1+x^2)^2} \right] \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \left[\frac{1}{2} \left(1 + \sqrt{\frac{x^2}{1+x^2}} \right) \right]^k \right\} \bigg|_{x=0} \\ &+ \left\{ N_c \left[\frac{1}{2} \left(1 - \sqrt{\frac{x^2}{1+x^2}} \right) \right]^{n-m+i} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \left[\frac{1}{2} \left(1 + \sqrt{\frac{x^2}{1+x^2}} \right) \right]^{k-1} \frac{1}{2} \frac{1}{2} \left(\frac{x^2}{1+x^2} \right)^{-1/2} \frac{2x}{(1+x^2)^2} \right\} \bigg|_{x=0} \end{aligned}$$

Here, the 1st item goes to

$$\begin{aligned}
& \left\{ \frac{N_c (n-m+i) \left(-\frac{1}{4}\right) \left(\frac{x^2}{1+x^2}\right)^{-1/2}}{2^{n-m+i-1}} \frac{2x}{(1+x^2)^2} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \frac{1}{2^k} \right\} \Bigg|_{x=0} \\
&= \left\{ N_c \frac{n-m+i}{2^{n-m+i-1}} \left(-\frac{1}{2}\right) \frac{1}{(1+x^2)^{3/2}} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \frac{1}{2^k} \right\} \Bigg|_{x=0} \\
&= \left\{ \frac{N_c (n-m+i) \left(-\frac{1}{4}\right) \left(\frac{x^2}{1+x^2}\right)^{-1/2}}{2^{n-m+i-1}} \frac{2x}{(1+x^2)^2} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \frac{1}{2^k} \right\} \Bigg|_{x=0} \\
&= \left\{ N_c \frac{n-m+i}{2^{n-m+i-1}} \left(-\frac{1}{2}\right) \frac{1}{(1+x^2)^{3/2}} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \frac{1}{2^k} \right\} \Bigg|_{x=0} \\
&= -N_c \frac{n-m+i}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \frac{1}{2^k} = -N_c \frac{n-m+i}{2} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \frac{1}{2^k} \frac{1}{2^{n-m+i-1}}
\end{aligned}$$

With the properties of the binomial coefficient, $\sum_{k=0}^n C_k^{n+k} [x^{n+1}(1-x)^k + (1-x)^{n+1}x^k] = 1$, we may

manipulate above term as

$$-N_c \frac{n-m+i}{2} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \left[\frac{1}{2^{n-m+i}} \cdot \left(1 - \frac{1}{2}\right)^k + \left(1 - \frac{1}{2}\right)^{n-m+i} \frac{1}{2^k} \right] = -N_c \frac{n-m+i}{2}$$

For the 2nd, when $x = 0$, there is

$$\begin{aligned}
& \frac{N_c}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} k \frac{1}{2^{k-1}} \left(1 + \sqrt{\frac{x^2}{1+x^2}}\right)^{k-1} \frac{1}{2} \frac{1}{(1+x^2)^{3/2}} \Bigg|_{x=0} \\
&= \frac{N_c}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} \binom{n-m+i-1+k}{k} \frac{k}{2^k}
\end{aligned}$$

Thus, the d_i turns to

$$d_i = -N_c \frac{n-m+i}{2} + \frac{N_c}{2^{n-m+i}} \sum_{k=0}^{n-m+i-1} C_{n-m+k+i-1}^k \frac{k}{2^k} \quad (\text{A6.2})$$

Appendix A7. Local Robustness δ_i'

Local robustness of the i th branch power allocation - δ_i' is given by

$$\delta_i' = \frac{\partial \overline{P_B} / \overline{P_B}}{\partial \alpha_i / \alpha_i} = \frac{\partial \overline{P_B}}{\partial \alpha_i} \cdot \frac{\alpha_i}{\overline{P_B}} = \lambda \frac{\alpha_i}{\overline{P_B}}$$

Because the Lagrange multiplier λ in (5.7) can be simplified as

$$\begin{aligned} \lambda &= \frac{1}{(4D_s\gamma_0)^{(n-m+1)}} \left(\frac{a_1}{m} \right)^{n-m+2} \left[1 - \frac{b_2}{mc_1^{n-m+3}\sqrt{4D_s\gamma_0}} \right]^{-(n-m+2)c_1} \\ &= \frac{1}{(4D_s\gamma_0)^{n-m+1}} \left(\frac{N_c^{n-m+2} \sqrt{N_c(n-m+1)C_{2(n-m+1)-1}^{n-m+1}}}{m} \right)^{n-m+2} \left[1 - \frac{a_2 a_1^{\frac{n-m+2}{n-m+3}} m^{\frac{c_1}{c_2}}}{m \frac{(n+1)!}{(n-m+1)!(n-m+2)} \sqrt{4D_s\gamma_0}} \right]^{\frac{(n+1)!}{(n-m+1)!}} \\ &\approx \frac{N_c(n-m+1) C_{2(n-m+1)-1}^{n-m+1}}{(4D_s\gamma_0)^{n-m+1} m^{n-m+2}} \end{aligned} \quad (A7.1)$$

and the BLER can be approximated to $\overline{P_{e,1}}$,

$$\overline{P_B} \approx \overline{P_{e,1}} = N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{(4\alpha_1 \cdot D_s\gamma_0)^{n-m+1}} \approx N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{(4m \cdot D_s\gamma_0)^{n-m+1}} \quad (A7.2)$$

therefore the robustness δ_1 according to α_1 goes to

$$\delta_1' = \lambda \frac{\alpha_1}{\overline{P_B}} \approx \frac{N_c(n-m+1) C_{2(n-m+1)-1}^{n-m+1}}{(4D_s\gamma_0)^{n-m+1} m^{n-m+2}} \frac{m}{N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{(4m \cdot D_s\gamma_0)^{n-m+1}}} = n-m+1 \quad (A7.3)$$

For the robustness δ_i according to α_i , it is generally given by

$$\begin{aligned} \delta_i' &= \lambda \frac{\alpha_i}{\overline{P_B}} \approx \left[\frac{N_c(n-m+1) C_{2(n-m+1)-1}^{n-m+1}}{(4D_s\gamma_0)^{n-m+1} m^{n-m+2}} \frac{b_i}{(4D_s\gamma_0)^{\frac{i-1}{n-m+i+1}}} \right] \frac{1}{\overline{P_B}} \\ &= \frac{N_c(n-m+1) C_{2(n-m+1)-1}^{n-m+1}}{(4D_s\gamma_0)^{n-m+1} m^{n-m+2}} \frac{b_i}{(4D_s\gamma_0)^{\frac{i-1}{n-m+i+1}}} \frac{1}{N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{(4m \cdot D_s\gamma_0)^{n-m+1}}} \\ &= \frac{(n-m+1)b_i}{m(4D_s\gamma_0)^{\frac{i-1}{n-m+i+1}}} = \frac{(n-m+1)}{m(4D_s\gamma_0)^{\frac{i-1}{n-m+i+1}}} \sqrt{\frac{(n-m+i)C_{2(n-m+i)-1}^{n-m+i} \cdot m^{n-m+2}}{(n-m+1)C_{2(n-m+1)-1}^{n-m+1}}} \end{aligned} \quad (A7.4)$$

Appendix A8. Truncated Asymptotic Approximation of $\bar{P}_B(\alpha)$

Given $\alpha = \alpha_{\text{opt}}$,

$$\begin{cases} \alpha_{1,\text{opt}} = m - \sum_{i=2}^m \alpha_{i,\text{opt}} \\ \alpha_{i,\text{opt}} \approx b_i (4D_s \gamma_0)^{(1-i)/(n-m+i+1)}, (i \geq 2) \end{cases}$$

and

$$b_i = \frac{(n-m+i)C_{2(n-m+i)-1}^{n-m+i} \cdot m^{n-m+2}}{(n-m+1)C_{2(n-m+1)-1}^{n-m+1}} \\ c_i = \frac{(n+1)!}{(n-m+1)!(n-m+i+1)}$$

the BLER approximation by keeping $\bar{P}_{e,i}$ of the first two steps is

$$\begin{aligned} \sum_{i=1}^m \bar{P}_{e,i}(\alpha_{\text{opt}}) &\approx N_c \sum_{i=1}^m \frac{C_{2(n-m+i)-1}^{n-m+i}}{(4\alpha_{i,\text{opt}} \cdot D_s \gamma_0)^{n-m+i}} \\ &\approx N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{\left\{4 \left[m - b_2 (4D_s \gamma_0)^{-\frac{1}{n-m+3}} \right] D_s \gamma_0 \right\}^{n-m+1}} + N_c \frac{C_{2(n-m+2)-1}^{n-m+2}}{\left\{4 \left[b_2 (4D_s \gamma_0)^{-\frac{1}{n-m+3}} \right] D_s \gamma_0 \right\}^{n-m+2}} \\ &= N_c \left[\frac{C_{2(n-m+1)-1}^{n-m+1}}{(4D_s \gamma_0)^{n-m+1}} \frac{1}{m^{n-m+1} - (n-m+1)m^{n-m} b_2 (4D_s \gamma_0)^{-\frac{1}{n-m+3}}} + \frac{C_{2(n-m+2)-1}^{n-m+2}}{(4D_s \gamma_0)^{n-m+2}} \frac{(4D_s \gamma_0)^{\frac{n-m+2}{n-m+3}}}{b_2^{n-m+2}} \right] \end{aligned}$$

Also because

$$\left(\frac{n-m+1}{m} \right)^2 b_2^2 (4D_s \gamma_0)^{-\frac{2}{n-m+3}} \ll 1$$

one may have the approximation of

$$\frac{1}{1 - \left(\frac{n-m+1}{m} \right) b_2 (4D_s \gamma_0)^{-\frac{1}{n-m+3}}} \approx 1 + \left(\frac{n-m+1}{m} \right) b_2 (4D_s \gamma_0)^{-\frac{1}{n-m+3}}$$

Thus, above $\bar{P}_B$ approximation goes to

$$\sum_{i=1}^m \overline{P_{e,i}}(\mathbf{a}_{\text{opt}}) \approx N_c \frac{C_{2(n-m+1)-1}^{n-m+1}}{(4D_s \gamma_0 m)^{n-m+1}} \cdot \left[\left(1 + \frac{n-m+1}{m} b_2 (4D_s \gamma_0)^{-\frac{1}{n-m+3}} \right) + \left(\frac{C_{2(n-m+2)-1}^{n-m+2} m^{n-m+1} (4D_s \gamma_0)^{-\frac{1}{n-m+3}}}{C_{2(n-m+1)-1}^{n-m+1} b_2^{n-m+2}} \right) \right]$$

Appendix B. Parallel Matlab® Toolkits

Appendix B1. Evaluation of Existent Parallel Matlab® Toolkits

1. “*Parmatlab*”

(<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=217&objectType=file>)

Category: Embarrassing parallel in an UNIX or Windows environment. Work on a Multi-processor or Network platform.

Status: Dormant. Support Matlab R12. The last update was 2001-04-25.

Test Result: Could not pass its self test with Matlab R2006b. (Error: “Opening port problem happens at mex file (tcpipmex.c), ‘Too many open port?’”)

Conclusion: The problem happens because of a low-level .c file which is not able to be fixed.

2. “*Matlab Parallelization Toolkit*”

(<http://einar.heiberg.se/misc.html#mpi>)

Category: Embarrassing parallel in an UNIX environment. Work on a Multi-processor or Network platform.

Status: Active. Support Matlab R11, R12, R13 and R14SP1.

Test Result: Has not been tried in a UNIX environment yet.

Conclusion: According to the documents offered by the developer, this tool has neat syntax especially for the calculation with low inter-process communication, e.g. the “for” loop. However, as an embarrassing parallel, it is still limited and needs multiple licenses for the group calculation.

3. ***“MULTICORE”***

(<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=13775&objectType=file>)

Category: Embarrassing parallel in a Windows environment. Work on a Multi-processor or Network platform.

Status: Active. Support R2006b

Test Result: Multi-processor mode did work, but the calculation speed was not improved notably. Therefore, its efficiency is code-depending.

Conclusion: With limited functions, this tool is still easy to use. No special toolboxes are required by it; neither compilation of mex-files is demanded. Nevertheless, it is not credible in the numerous iterations scenario, where single iteration only needs few seconds to execute and the overhead of collection and distribution is significant.

4. ***“Cornell Multitasking Toolbox for Matlab (CMTM)”***

(<http://www.tc.cornell.edu/services/support/forms/cmtm.p2.htm>)

Category: Message passing parallel in an UNIX or Windows environment. Work on a Multi-processor or Network platform.

Status: Active but a time-limited trail version. Support R2006b.

Test Result: Additional MPI/Pro 1.7.0 tools box is required.

Conclusion: N/A

5. ***“MatlabMPI”***

(<http://www.ll.mit.edu/MatlabMPI/>)

Category: Message passing parallel in an UNIX or Windows environment. Work on a Multi-processor or Network platform.

Status: Dormant.

Test Result: Multi-processor mode works. By one simple MatlabMPI command, tested codes completed their calculations within 30%-40% less time on a hyper threading 2-core PC. As the multi-processor mode of this tool is just a command with the original code name as one of its inputs, its efficiency is code-dependent. Further, the networking mode could not work, because MatlabMPI can only recognize a remote computer name defined by UNIX syntax. Neither Windows PC names nor IP addresses are acceptable.

Conclusion: Calculation efficient is code-dependent. Resource arrangement is transparent and uncontrollable to users.

6. ***“pMatlab”***

(<http://www.ll.mit.edu/pMatlab/>)

Category: Message passing parallel in an UNIX or Windows environment. Work on a Multi-processor or Network platform.

Status: Active. Upgraded version of “MatlabMPI”

Test Result: Similar program syntax to the original Matlab.

1) Much more flexible parallel commands for distribution and collection operations than commands in MatlabMPI; 2) Underlying new data type - “distributed matrix”, which maps grids data to different processors; 3) The same problem in its networking mode as in MatlabMPI.

Conclusion: Innovative “parallel array programming” mechanism is introduced. More flexible and powerful parallel calculation applications can be developed by this kit than by MatlabMPI.

7. “Netslove”

(<http://icl.cs.utk.edu/netsolve/index.html>)

Category: Backend support parallel on a Network platform. Host works in an UNIX environment only. Slaveries work in an UNIX or Windows environment.

Status: Active

Test Result: Could not pass its self-test if there is not any available UNIX host server. For the default processing server, “netsolve.cs.utk.edu”, there were more than 13,000 tasks in its queue.

Conclusion: A local UNIX host serve is needed to be setup to accept and distribute tasks over a heterogeneous network.

8. “Distributed Computing Toolbox 3.1” from Mathworks™

(<http://www.mathworks.com/products/distribtb/>)

Category: Message passing parallel in a Windows environment Work on a Multi-processor or Network platform. For the Network mode, an additional “MATLAB Distributed Computing Engine” is needed.

Status: Active

Test Result: 30-days trial license is acquirable from Mathworks™. The “pmode” works on Matlab® R2007. Almost same calculation efficiency improvement achieved on a dual-core PC as that achieved by pMatlab. (See Appendix10 for the usage details.)

Conclusion: Commercial software. The most comprehensive tools package offering both data and task parallelism in all candidates. Well documentary / technical supported.

9. ***“Start-P” originate from “MATLAB*P”*** by MIT – Ron Choy and etl.

(<http://www.interactivesupercomputing.com/products/starpandmatlab.php/>)

Category: Message passing parallel in an UNIX or Windows environment.

Status: Active

Test Result: N/A

Conclusion: Commercial software.

Appendix B2. Guide of Using pMatlab

1. Installation of pMatlab Toolbox

- Unpack pMatlab software package and copy the fold of the toolbox under Matlab directory;
- Go to C:\matlab\r2006b\toolbox\local\startupsav and add below lines in and rename the file as “startup.”

```
addpath C:\matlab\r2006b\pMatlab\src;
addpath C:\matlab\r2006b\pMatlab\MatlabMPI\src;
addpath C:\matlab\r2006b\pMatlab\examples;
addpath C:\matlab\r2006b\pMatlab\MatlabMPI\examples;
warning off;
addpath .\MatMPI;
pause(2.0);
warning on;
```

- Open a Matlab working space and type “RUN”. If the testing script is executed successfully, it means pMatlab was installed properly and parallel calculation can then be testified in this Matlab environment.

2. Parallel Simulation Conduction

Below shows a simple example of transmission error estimation by parallel computation with pMatlab. The sentences between the red arrows implement the data distribution and aggregation among the processors.

```
pMatlab_Init; %Initialize pMatlab
N_error=0; %Number of transit errors
%Distributed matrix constructed on multi-processors under the
instruction of map
map1=map([1 2].{ }.0:1); %Create "Map"
%Generate source data string and map it on local processors
d=round(rand(1,500,map1));
%All parallel calculations are implemented upon distributed matrix
(dmat)
x_d=2*d-1; %transmitter output with anti-podal data string
n_d=0.4*rand(1,500,map1); %Generate AWG noise with std dev of 0.4
y_d=x_d+n_d; %Receiver input
est_d=round(y_d(:, :))+1; %Estimated data
error=(est_d==d); %Error dmat
%Return regular double error matrix as the aggregation of error dmat
error1=agg(error);
%All parallel calculation results are assembled by "aggregation. Data
processing based on regular matrix.
for j=1:500
    if error1(1,j)==1
        N_error=N_error+1;
    end;
end;
disp('SUCCESS');
pMatlab_Finalize;
```

Now, suppose we have one Matlab® function defined by a .m file – “c9_MCBPSKrun.m” – estimating BER of a BPSK signal over an AWGN channel, the according parallel calculation version can be achieved after following manipulations.

- Test c9_MCBPSKrun.m and make sure with certain inputs – “N”, the number of MC iterations and “snrdB”, the SNR of signal – of the function give a correct “BER” return.
- Manipulate the inputs and output of function c9_MCBPSKrun as below

```
pMatlab_Init; %Initialize pMatlab
Ncpus=pMATLAB.comm_size; %cpus numbers from pMATLAB internal
parameter "pMATLAB.comm_size"
map1=map([Ncpus 1], {}, 0:Ncpus-1); %Create "Map" - the
distribution matrix data structure
BER=zeros(Ncpus,1,map1); %Create output distribution matrix
my_BER=local(BER); %Define the local portion of the
output distributed matrix
[my_BER]=c9_MCBPSKrun(N/Ncpus,snrdB); %Local simulation
running, N/Ncpus is the number of the iterations on any single
processor
BER=put_local(BER,my_BER); %Assemble local portions of output data
BER_final=agg(BER); %Aggregate distributed matrix to a
regular matrix
avg_BER=sum(BER_final)/Ncpus; %Average results from different
cpus
```

- Save above script with a name, say “pMCBPSKrun.m” in either
“C:\matlab\r2006b\work” or “C:\matlab\r2006b\pMatlab\examples”.
- Execute below pMatlab command to start the parallel execution of the script
“pMCBPSKrun” as

```
eval(MPI_Run('pMCBPSKrun', 2, {})); % Where "2" means the
number of CPUs and "{}" means on local PC
```

- Clean up the temporary files and folds generated by parallel running with

```
MPI_Abort;                                % Abort left over jobs
pause(2.0);
MatMPI_Delete_all;                        % Delete left over MPI directory
pause(2.0);
```

- Check the output “avg_BER”. It will be find that the estimation result is equal to the result from regular calculation “[BER]=c9_MCBPSKrun(N,snrdB);” but the running time is roughly N_{cpus} -fold cut.

3. Useful Web Links and Guidelines

- Main webpage of “pMatlab” (<http://www.ll.mit.edu/pMatlab/>) with basic installation / running instructions for first time user
- Good brief introduction of how to programming with pMatlab
(http://www.ll.mit.edu/pMatlab/files/pMatlab_param_sweep.pdf)
- Page 13 to 17 of the “Introduction of pMatlab V0.7” helping understanding the underlying new type of data structures.
(http://www.ll.mit.edu/pMatlab/files/pMatlab_intro.pdf)
- Manu of pMatlab commands
(http://www.ll.mit.edu/pMatlab/files/pMatlab_v0.7_func_ref.pdf)

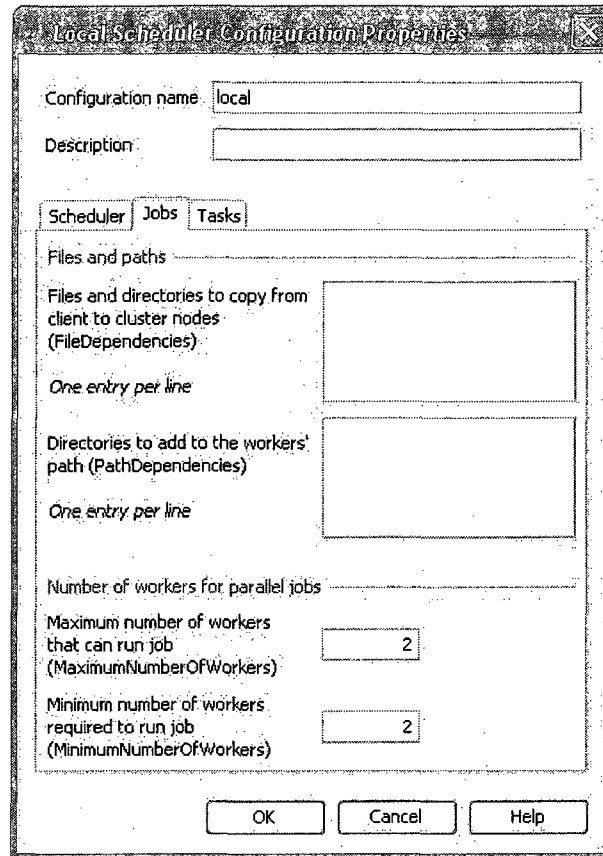
Appendix B3. Usage of the Parallel Computing Tools (PCT) for Matlab 2007b

1. Configure local scheduler

Define number of local “workers” for parallel execution through

“Distributed” (main menu) → “Manage configuration” → “local” → “property” → “jobs”

Set the “maximum / minimum number of workers” to the number of local CPUs



The image shows a Windows-style dialog box titled "Local Scheduler Configuration Properties". It has a close button (X) in the top right corner. The dialog is divided into three tabs: "Scheduler", "Jobs", and "Tasks". The "Scheduler" tab is currently selected. Inside the "Scheduler" tab, there are several sections:

- Configuration name:** A text box containing the word "local".
- Description:** An empty text box.
- Files and paths:** A section containing two text boxes:
 - Files and directories to copy from client to cluster nodes (FileDependencies):** An empty text box.
 - Directories to add to the workers' path (PathDependencies):** An empty text box.
- Number of workers for parallel jobs:** A section containing two text boxes:
 - Maximum number of workers that can run job (MaximumNumberOfWorkers):** A text box containing the number "2".
 - Minimum number of workers required to run job (MinimumNumberOfWorkers):** A text box containing the number "2".

At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

2. Start P-Mode

Type “pmode start” or “pmode start numlabs” in command window. PMODE window then is initialized.

```

tic;[n,pi]=c9_EstimatePi_2(2,1e3);toc;
n
pi
tic;[n,pi]=c9_EstimatePi_1(2,1e7);toc;
n
pi
tic;[BER]=c10_MCQPSKdelay_1(1e2);toc;
tic;[BER]=c10_MCQPSKdelay_1(2,1e2);toc;
tic;[n,BER]=c9_MCBPSKrun(2,1e6,3);toc;
tic;[n,BER]=c9_MCBPSKrun_1(2,1e6,3);toc;
tic;[BER]=c10_MCQPSKdelay_1(2,1e2);toc;
tic;[n,BER]=c9_MCBPSKrun_1(2,1e7,3);toc;
BER
n
tic;[n,BER]=c9_MCBPSKrun_1(2,1e6,3);toc;
BER
tic;[n,BER]=c9_MCBPSKrun_1(2,1e6,3);toc;
n
BER
tic;[BER]=c10_MCQPSKdelay_1(2,1e2);toc;
n
tic;[n,BER]=c10_MCQPSKdelay_1(2,1e2);toc;
n
BER
%-- 20/05/08 2:03 PM --%
P>>

```

3. Re-code MC simulations

The original code:

```

function[n,BER]=c9_MCBPSKrun(IterationNumber,snrdB)
N=IterationNumber;           %Number of iterations
snr=10.^(snrdB/10);          %Convert from dB to linear unit
NoiseSigma=sqrt(1/(2*snr));   %Scale noise level
Errors=0;
n=0;
for k=1:N
    %Initialize transmitted symbols
    n=n+1;
    [TxOutput,TxSymbols]=random_binary(1,1);
    RxInput=zeros(1,1);

```

```

    IntegratorOutput=zeros(1,1);
    DetectedSymbols=zeros(1,1);
    RxInput=TxOutput+NoiseSigma*randn(1);
    %Pass Received signal through matched filter and count errors
    IntegratorOutput=filter(1,1,RxInput);
    DetectedSymbols=(1-sign(IntegratorOutput))/2;
        if (DetectedSymbols~=TxSymbols)
            Errors=Errors+1;
        end;
end;
%Sum up total errors in each "batch" and calculate BER
BER=Errors/N;
end
where
function [x, bits]=random_binary(nbits, nsamples)
%This function genrates a random binary waveform of length n bits
%Sampled at a rate of nsamples/bit.
x=zeros(1, nbits*nsamples);
bits=round(rand(1, nbits));
for m=1:nbits;
    for n=1:nsamples
        index=(m-1)*nsamples+n;
        x(1, index)=(-1)^bits(m);
    end;
end;
%End
end

```

Modified to

```

function[n, BER]=c9_MCBPSKrun_1(Num_labs, IterationNumber, snrdB)
N=IterationNumber; %Number of iterations
snr=10.^(snrdB/10); %Convert from dB to linear unit
NoiseSigma=sqrt(1/(2*snr)); %Scale noise level
Errors=0;
n=0;
for k=drange(1:Num_labs) %Set up parallel for loops
    for kk=1:N/Num_labs %Num labs-number of local
        if (DetectedSymbols~=TxSymbols)

```

```

        Errors=Errors+1;
    end;
end;
end;
%Sum up total errors in each "batch" and calculate BER
BER=Errors/(N/Num_labs);
end
function [x, bits]=random_binary(nbits, nsamples)
end

```

4. *Execute Parallel For-Loop*

In PMODE command window, type in

```
"tic; [n, BER]=c9_MCBPSKrun_1(2, 1e6, 3); toc;" .
```

returns

```
"Elapsed time is 16.401616 seconds" and "Elapsed time is 16.842230 seconds."
```

and

```
"n=500000" in both separate PMODE interactive windows.
```

```
"BER=0.0228" and "BER=0.0228"
```

(Note: Above results need to be averaged to get the final answers)

5. *Compare with Original For-Loop*

In Matlab command window, type in

```
"tic; [n, BER]=c9_MCBPSKrun(1e6, 3); toc;" Returns
```

```
"Elapsed time is 33.879169 seconds."
```

and

```
"n=1000000" and "BER=0.0229"
```

6. *Quit PMODE*

In Matlab command window, type in “`pmode exit`”

7. *Advantages of the PMODE*

- Once PMODE is initialed, no more initialization time for every single simulation execution.
- Easy re-coding method.

Appendix C. Matlab® Code

Appendix C1. Functions

Exact Instantaneous Ps

```
function out=ExaInsPsTol(Modulation)
%Generating Instantaneous Ps for
%the whole gammadB range (0~18dB)

if Modulation == 'QPSK'
    M=4;
else if Modulation == 'MPSK'
    M=8;
    else if Modulation == 'SQAM'
        M=8;
        else if Modulation == 'MQAM'
            M=16;
        end;
    end;
end;

end;
gammadB=0:3:18;
for j=1:length(gammadB)
    gammaw=10^(gammadB(j)/10);
    %analytical avg SER of modulations with n=MRC
    if Modulation == 'QPSK'|Modulation == 'MPSK'
        avgPe=1/pi*quad(@(phi)MPSK(M,gammaw,phi),1e-10,(M-1)*pi/M);
    else if Modulation == 'SQAM'|Modulation == 'MQAM'
        z1=4/pi*(1-1/sqrt(M))*quad(@(phi)MQAM(M,gammaw,phi),1e-10,0.5*pi);
        z2=4/pi*(1-1/sqrt(M))^2*quad(@(phi)MQAM(M,gammaw,phi),1e-10,0.25*pi);
        avgPe=z1-z2;
    end;
    end;
    out(j)=avgPe;
end;
end

function out=MPSK(M,gamma,phi)
out=exp(-gamma*sin(pi/M)^2./sin(phi).^2);
end

function out=MQAM(M,gamma,phi)
%Moment Generating Function of M-QAM
*L-diversity order
%phi-integral variance in Q function
out=exp(-gamma*3./(2*(M-1).*sin(phi).^2));
end
```

Closed Form Instantaneous Ps

```
function [out1,out2]=CFInsPsTol(Modulation)
%out1-asymptotic approx (Rappaport-6.127); out2-the union bound;
if Modulation == 'QPSK'
    M=4;
else if Modulation == 'MPSK'
    M=8;
    else if Modulation == 'SQAM'
        M=8;
    end;
end;
```



```

%Substituting union Q-function bound into
%the closed-form SER expression of BPSK over single Rayleigh
%- Proakis 14.3-7
%out1-general approx (Rappaport-6.127); out2-the simplified union bound;
if Modulation == 'QPSK'
    M=4;
    B1=2;B2=B1;
    A1=sin(pi/M)^2;A2=A1;
else if Modulation=='MPSK'
    M=8;
    B1=2;B2=B1;
    A1=sin(pi/M)^2;A2=A1;
else if Modulation == 'SQAM'
    M=8;
    B1=4*(1-1/sqrt(M));
    A1=3/(M-1)/2;
    B2=9/4;
    A2=1/5;
else if Modulation == 'MQAM'
    M=16;
    B1=4*(1-1/sqrt(M));
    A1=3/(M-1)/2;
    B2=3;
    A2=1/10;
end;
end;
end;
end;
gammasB=0:3:39;
for j=1:length(gammasB)
    gammaW=10^(gammasB(j)/10);
    mu1=sqrt(A1*gammaW/(1+A1*gammaW));
    mu2=sqrt(A2*gammaW/(1+A2*gammaW));
    out1(j)=B1/2*(1-mu1);
    out2(j)=B2/2*(1-mu2);
end
end

```

Exact Average Ps with Diversity Combining

```

function out=ExaAvgPsTol(n,Modulation)
%Generating AvgPs for n-i.i.d. Rayleigh fading
%The whole gammasB range (0~39dB)

if Modulation == 'QPSK'
    M=4;
else if Modulation == 'MPSK'
    M=8;
else if Modulation == 'SQAM'
    M=8;
else if Modulation == 'MQAM'
    M=16;
end;
end;
end;
end;
gammasB=0:3:39;
for j=1:length(gammasB)
    gammaw=10^(gammasB(j)/10);
    %analytical avg SER of modulations with n=MRC
    if Modulation == 'QPSK'|Modulation == 'MPSK'

```

```

        out(j)=1/pi*quad(@(phi)MGFMPSK(M,gammaw,n,phi),1e-10,(M-1)*pi/M);
    else if Modulation == '8QAM'|Modulation == 'MQAM'
        z1=4/pi*(1-1/sqrt(M))*quad(@(phi)MGFMQAM(M,gammaw,n,phi),1e-10,0.5*pi);
        z2=4/pi*(1-1/sqrt(M))^2*quad(@(phi)MGFMQAM(M,gammaw,n,phi),1e-10,0.25*pi);
        out(j)=z1-z2;
    end;
end;
end;
end;

function out=MGFMPSK(M,gamma,L,phi)
%Moment Generating Function of M-PSK
%L-diversity order
%phi-integral variance in Q function
out=(1+gamma.*(sin(pi/M)^2)./(sin(phi).^2)).^(-L);
end

function out=MGFMQAM(M,gamma,L,phi)
%Moment Generating Function of M-QAM
%L-diversity order
%phi-integral variance in Q function
out=(1+3*gamma./(2*(M-1).*sin(phi).^2)).^(-L);
end

```

Closed Form Average P_s with Diversity Combining

```

function [out1,out2]=CFInsPsTol(Modulation)
%out1-asymptotic approx (Rappaport-6.127); out2-the union bound;
if Modulation == 'QPSK'
    M=4;
else if Modulation == 'MPSK'
    M=8;
    else if Modulation == '8QAM'
        M=8;
        else if Modulation == 'MQAM'
            M=16;
        end;
    end;
end;
end;
gammadB=0:3:18;
for j=1:length(gammadB)
    gammaw=10^(gammadB(j)/10);
    %analytical avg SER of modulations with n=MRC
    if Modulation == 'QPSK'|Modulation == 'MPSK'
        out1(j)=2*qfunc(sqrt(2*gammaw)*sin(pi/M));
        out2(j)=out1(j);
    else if Modulation == '8QAM'
        out2(j)=9/4*qfunc(sqrt(2*gammaw/5));
    else if Modulation == 'MQAM'
        out2(j)=3*qfunc(sqrt(gammaw/5));
    end;
    end;
    out1(j)=4*(1-1/sqrt(M))*qfunc(sqrt(3/(M-1)*gammaw));
end;
end;

```

Asymptotic Approximate Average P_s with Diversity Combining

```

function [out3,out4]=AsympAvgPsTol(n,Modulation)
%From the general diversity / coding gain model in Wang and Giannakis's

```

```

% "A Simple and General Parameterization Quantifying Performance
% in Fading Channel"
% out3-corresponding to(Rappaport-6.127); out4-corresponding to union bound;
if Modulation == 'QPSK'
    M=4;
    B1=2;B2=B1;
    A1=sin(pi/M)^2;A2=A1;
else if Modulation=='MPSK'
    M=8;
    B1=2;B2=B1;
    A1=sin(pi/M)^2;A2=A1;
else if Modulation == 'SQAM'
    M=8;
    B1=4*(1-1/sqrt(M));
    A1=3/(M-1)/2;
    B2=9/4;
    A2=1/5;
else if Modulation == 'MQAM'
    M=16;
    B1=4*(1-1/sqrt(M));
    A1=3/(M-1)/2;
    B2=3;
    A2=1/10;
end;
end;
end;

gammadB=0:3:39;
for j=1:length(gammadB)
    gammaw=10^(gammadB(j)/10);
    out3(j)=B1/2/sqrt(pi)*(1/A1/gammaw)^n*gamma(n+0.5)/gamma(n+1);
    out4(j)=B2/2/sqrt(pi)*(1/A2/gammaw)^n*gamma(n+0.5)/gamma(n+1);
end;

```

Numerical Optimum Alpha α_{opt} Derivation

```

function alpha=Num_Alpha(m,n,Modulation)
%Finding optimum power allocation numerically
warning('off');
gammadB=-27:3:60;
alpha=zeros(length(gammadB),m);
alpha0=zeros(1,m);
Aeq=ones(1,m);
beq=m;
lb=zeros(1,m);
options=optimset('LargeScale','off','TolX',1e-010,'TolFun',1e-012,'TolCon',1e-012,'Display','off');
for j=1:length(gammadB)
    if Modulation == 'BPSK'|Modulation == 'QPSK'|Modulation == 'MPSK'|Modulation == 'MQAM'
        alpha(j,:)=fmincon(@(alpha)
log(ExaAvgBLER(gammadB(j),m,n,alpha,Modulation)),alpha0,[],[],Aeq,beq,lb,[],[],options);
    else if Modulation == 'SQAM'
        alpha(j,:)=fmincon(@(alpha)
log(ExaAvgBLER(gammadB(j),m,n,alpha,Modulation)),alpha0,[],[],Aeq,beq,lb,[],[],options);
    end;
end;
end;
alpha=alpha';

```

end

Approximate Optimum Alpha α_{opt}

```
function [alphaz,alpha]=App_Alpha(m,n,Modulation)
%Finding close-form optimum alpha coefficient in an m by n VELAST system
%alpha-vector of power allocation coefficient
%alphaz-vector of power allocation coefficient around SNR=0
%m-number of Tx antennas
%n-number of Rx antennas
%gammadB-average SNR in dB
%q,k-modulation dependent parameters, Ins.SER-q*Q(sqrt(k*gamma))

%define modulation specifications
gammadB=0:3:39;
if Modulation == 'BPSK'
    A=1; B=1;
else if Modulation == 'QPSK'
    M=4;
    B=2;
    A=sin(pi/M)^2;
else if Modulation=='MPSK'
    M=8;
    B=2;
    A=sin(pi/M)^2;
else if Modulation == '8QAM'
    M=8;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
else if Modulation == 'MQAM'
    M=16;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
end;
end;
end;
end;

%calculate the alpha(j,:) for SNR=0
for j=1:m
    d=d(m,n,B);
    alphaz(j)=d(j)^2*m/sum(d.^2);
end;

%calculate the alpha(j,:) for large SNR, the power coefficient vector for
%the jth Tx branch
for kk=1:length(gammadB)
    gammaW=10^(gammadB(kk)/10);
    for j=1:m
        alpha(j,kk)=b(m,n,j)*(4*A*gammaW)^(-(j-1)/(n-m+j+1))*(1-
        b(m,n,2)/(m*c(m,n,1)*(4*A*gammaW)^(1/(n-m+3))))^c(m,n,j);
    end;
%normalize alpha(j,kk)
    alpha(:,kk)=m*alpha(:,kk)/sum(alpha(:,kk));
    alpha(1,kk)=m-sum(alpha(2:m,kk));
end;
end

function c=c(m,n,j)
%return optimum parameter c(j) for the jth step
```

```

c=prod(n-m+1+1:n+1)/(n-m+j+1);
end

function b=b(m,n,j)
%return optimum parameter b(j) for the jth step
b=((n-m+j)*nchoosek(2*(n-m+j)-1,n-m+j)*m^(n-m+2)/(n-m+1)/nchoosek(2*(n-m+1)-1,n-m+1))^(1/(n-m+j+1));
end

function d=d(m,n,B)
%return optimum parameter d
for i=1:m
    sum=0;
    for k=0:(n-m+i-1)
        sum=sum+nchoosek(n-m+k+i-1,k)*k/2^k;
    end;
    d(i)=B*(-(n-m+i)/2+1/2^(n-m+i)*sum);
end;
end

```

Exact Average BLER in ZF-VBLAST

```

function out=ExaAvgBLERTol(m,n,alpha,Modulation)
%Different from function 'ExaAvgPb', where BLER with certain SNR is found
%in this function, AvgPb for
%the whole gammadB range (0~39dB) are generated
%m-number of Tx antennas
%n-number of Rx antennas

gammadb=0:3:39;
for j=1:length(gammadb)
    out(j)=ExaAvgBLER(gammadb(j),m,n,alpha(:,j),Modulation);
end;
end
function out=ExaAvgBLER(gammadb,m,n,alpha,Modulation)
%Find Exact Avg. BLER for certain avg. SNR using Moment Generating Function
%(MGF) from Simon's (9.15) and (9.21)
%Lognormal final Exact Pb for tracking the optimum power allocation.
if Modulation == 'QPSK'
    M=4;
else if Modulation == 'MPSK'
    M=8;
    else if Modulation == 'QAM'
        M=8;
        else if Modulation == 'MQAM'
            M=16;
        end;
    end;
end;
end;

gammaw=10^(gammadb/10);
if Modulation == 'BPSK'
    for j = 1:m
        mu=sqrt(alpha(j)*gammaw/(1+alpha(j)*gammaw));
        P=0;
        for k=0:n-m+j-1
            P = P + nchoosek(n-m+j-1+k,k)*((1+mu)/2)^k;
        end;
        avgPe(j)=((1-mu)/2)^(n-m+j)*P;
    end;
else if Modulation == 'QPSK'|Modulation == 'MPSK'

```

```

        for j=1:m
            avgPe(j)=1/pi*quad(@(phi)MGFMPSK(M,alpha(j)*gammaw,n-m+j,phi),1e-10,(M-1)*pi/M);
        end;
    else if Modulation == 'SQAM'|Modulation == 'MQAM'
        for j=1:m
            z1=4/pi*(1-1/sqrt(M))*quad(@(phi)MGFMQAM(M,alpha(j)*gammaw,n-m+j,phi),1e-10,0.5*pi);
            z2=4/pi*(1-1/sqrt(M))^2*quad(@(phi)MGFMQAM(M,alpha(j)*gammaw,n-m+j,phi),1e-10,0.25*pi);
            avgPe(j)=z1-z2;
        end;
    end;
end;
end;
P=1;
for j=1:m
    P=P*(1-avgPe(j));
end;
out=1-P;
end

function out=MGFMPSK(M,gamma,L,phi)
out=(1+gamma*sin(pi/M)^2./(sin(phi).^2)).^(-L);
end

function out=MGFMQAM(M,gamma,L,phi)
out=(1+3*gamma./(2*(M-1).*sin(phi).^2)).^(-L);
end

```

Average BLER in ZF-VBLAST vs. α

```

function out=AvgBLERvsAlpha(m,n,gammasB,Modulation)
%Output a matrix with each AvgBLER element depending on alpha1 and alpha2
%in a optimum power allocated 3 by 3 VBLAST

if Modulation == 'QPSK'
    M=4;
else if Modulation == 'MPSK'
    M=8;
else if Modulation == 'SQAM'
    M=8;
else if Modulation == 'MQAM'
    M=16;
end;
end;
end;
gammaw=round(10^(gammasB/10));
%alpha(1) from 0.1 to 3.0
if m==3
    for jj=1:(m/0.1)
        alpha(1)=jj*0.1;
%alpha(2) from 0.1 to (m-alpha(1)-0.1), the last 0.1 leaving for alpha(3)
        for kk=1:((m-alpha(1))/0.1)-1
            alpha(2)=kk*0.1;
            alpha(3)=m-alpha(1)-alpha(2);
%analytical avg BLEP of modulations
            if Modulation == 'BPSK'
                for j = 1:m
                    mu=sqrt(alpha(j)*gammaw/(1+alpha(j)*gammaw));

```

```

        P=0;
        for k=0:n-m+j-1
            P = P + nchoosek(n-m+j-1+k,k)*((1+mu)/2)^k;
        end;
        avgPe(j)=((1-mu)/2)^(n-m+j)*P;
    end;
else
    %AvgPet(j)-the integral of MomentGeneratingFunction product of n-m+j MRC
    if Modulation == 'QPSK'|Modulation == 'MPSK'
        for j=1:m
            avgPe(j)=1/pi*quad(@(phi)MGFMPSK(M,alpha(j)*gammaw,n-
m+j,phi),1e-10,(M-1)*pi/M);
        end;
    else
        if Modulation == 'SQAM'|Modulation == 'MQAM'
            for j=1:m
                z1=4/pi*(1-
1/sqrt(M))*quad(@(phi)MGFMQAM(M,alpha(j)*gammaw,n-m+j,phi),1e-10,0.5*pi);
                z2=4/pi*(1-
1/sqrt(M))^2*quad(@(phi)MGFMQAM(M,alpha(j)*gammaw,n-m+j,phi),1e-10,0.25*pi);
                avgPe(j)=z1-z2;
            end;
        end;
    end;
    end;
    %avgELER-out(jj,kk)=1-prod(1-avgPe(j))
    P=1;
    for j=1:m
        P=P*(1-avgPe(j));
    end;
    out(jj,kk)=1-P;
end;
end;
else if m==2
    for jj=1:(m/0.1)
        alpha(1)=jj*0.1;
        %alpha(2)= m-alpha(1)-0.1
        alpha(2)=m-alpha(1);
        %analytical avg ELER of modulations
        if Modulation == 'BPSK'
            for j = 1:m
                mu=sqrt(alpha(j)*gammaw/(1+alpha(j)*gammaw));
                P=0;
                for k=0:n-m+j-1
                    P = P + nchoosek(n-m+j-1+k,k)*((1+mu)/2)^k;
                end;
                avgPe(j)=((1-mu)/2)^(n-m+j)*P;
            end;
        else
            %AvgPet(j)-the integral of MomentGeneratingFunction product of n-m+j MRC
            if Modulation == 'QPSK'|Modulation == 'MPSK'
                for j=1:m
                    avgPe(j)=1/pi*quad(@(phi)MGFMPSK(M,alpha(j)*gammaw,n-m+j,phi),1e-
10,(M-1)*pi/M);
                end;
            else
                if Modulation == 'SQAM'|Modulation == 'MQAM'
                    for j=1:m
                        z1=4/pi*(1-1/sqrt(M))*quad(@(phi)MGFMQAM(M,alpha(j)*gammaw,n-
m+j,phi),1e-10,0.5*pi);
                        z2=4/pi*(1-1/sqrt(M))^2*quad(@(phi)MGFMQAM(M,alpha(j)*gammaw,n-
m+j,phi),1e-10,0.25*pi);
                        avgPe(j)=z1-z2;
                    end;
                end;
            end;
        end;
    end;
end;
end;

```

```

        end;
    end;
end;
end;
%avgBLEK-out(jj, kk)=1-prod(1-avgPe(j))
P=1;
for j=1:m
    P=P*(1-avgPe(j));
end;
out(jj)=1-P;
end;
end;
end;

function out=MGFMPSK(M, gamma, L, phi)
%Moment Generating Function of QPSK
%L-diversity order
%phi-integral variance in Q function
out=(1+gamma*sin(pi/M)^2./(sin(phi).^2)).^(-L);
end

function out=MGFMQAM(M, gamma, L, phi)
%Moment Generating Function of 8QAM
%L-diversity order
%phi-integral variance in Q function
out=(1+3*gamma./(2*(M-1).*sin(phi).^2)).^(-L);
end

```

Exact λ

```

function lambda=ExaLambda(m,n,alpha,Modulation)
%Finding exact optimum power allocation parameter lambda derived from
%numerical alpha(1) found by the 'fmin' search.
%m-number of Tx antennas
%n-number of Rx antennas
%alpha-exact optimum power allocation
%gammadb-average SNR in dB

gammadb=0:3:39;
if Modulation == 'BPSK'
    A=1; B=1;
else if Modulation == 'QPSK'
    M=4;
    B=2;
    A=sin(pi/M)^2;
else if Modulation=='MPSK'
    M=8;
    B=2;
    A=sin(pi/M)^2;
else if Modulation == '8QAM'
    M=8;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
else if Modulation == '16QAM'
    M=16;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
end;
end;
end;

```

```

        end;
    end;

    %extract lambda from numerical alpha(i)-i=1
    i=1;
    alphai=alpha(i,:);
    for kk=1:length(gammadB)
        gammaW=10^(gammadB(kk)/10);
        lambda(kk)=B*(n-m+i)*nchoosek(2*(n-m+i)-1,n-m+i)/alphai(kk)^(n-
m+i+1)/(4*A*gammaW)^(n-m+i);
    end;
end

```

Approximate λ

```

function [lambdal,lambda2]=AppLambda(m,n,Modulation)
%Finding close form optimum power allocation parameter lambdal (5.6);
%Further approx. (6.2) lambda2
%alpha(j) can be expressed by single lambda
%m-number of Tx antennas
%n-number of Rx antennas
%gammadB-average SNR in dB

%define modulation specifications
gammadB=0:3:39;
if Modulation == 'BPSK'
    A=1; B=1;
else if Modulation == 'QPSK'
    M=4;
    B=2;
    A=sin(pi/M)^2;
else if Modulation=='MPSK'
    M=8;
    B=2;
    A=sin(pi/M)^2;
else if Modulation == 'SQAM'
    M=8;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
else if Modulation == 'MQAM'
    M=16;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
end;
end;
end;
end;

%approximation of lambda
for kk=1:length(gammadB)
    gammaW=10^(gammadB(kk)/10);
    a1=(B*(n-m+1)*nchoosek(2*(n-m+1)-1,n-m+1))^(1/(n-m+2));
    lambdal(kk)=(a1/m)^(n-m+2)/(4*A*gammaW)^(n-m+1)*(1-
b(m,n,2)/(m*c(m,n,1)*(4*A*gammaW)^(1/(n-m+3))))^(-(n-m+2)*c(m,n,1));
    lambda2(kk)=(a1/m)^(n-m+2)/(4*A*gammaW)^(n-m+1);
end;
end

function c=c(m,n,j)
%return optimum parameter c(j) for the jth step

```

```

c=prod(n-m+1+1:n+1)/(n-m+j+1);
end

function b=b(m,n,j)
%return optimum parameter b(j) for the jth step
b=((n-m+j)*nchoosek(2*(n-m+j)-1,n-m+j)*m^(n-m+2)/(n-m+1)/nchoosek(2*(n-m+1)-1,n-
m+1))^(1/(n-m+j+1));
end

```

SNR Gain

```

function [gain_num_opt,gain_app_opt,gain_cf]=SNRgain(m,n,alpha,Modulation)
%Finding both SNR gains corresponding numerical and approx alphas
% m-number of Tx antennas
% n-number of Rx antennas

options=optimset('LargeScale','off','TolX',1e-020,'TolFun',1e-024,'TolCon',1e-
024,'Display','off');
gammadB=0:3:39;
if Modulation == 'BPSK'
    A=1; B=1;
else if Modulation == 'QPSK'
    M=4;
    B=2;
    A=sin(pi/M)^2;
else if Modulation=='MPSK'
    M=8;
    B=2;
    A=sin(pi/M)^2;
else if Modulation == 'SQAM'
    M=8;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
else if Modulation == 'MQAM'
    M=16;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
end;
end;
end;
end;
end;
% 'Num_Alpha' function-numerical alpha by 'fmin' searching
num_alpha=alpha;

% 'App_Alpha' function-approx alpha from close form analytical expression
app_alpha=App_Alpha(m,n,Modulation);
% adjust approx alpha if alpha(1)<1
% generate SNR gain corresponding exact and approx alphas
for j=1:length(gammadB)
    if app_alpha(1,j)<1
        app_alpha(:,j)=1;
    end;
    gammaw=10^(gammadB(j)/10);
    exaPb_num_opt(j)=ExaAvgBLER(gammadB(j),m,n,num_alpha(:,j),Modulation);
    exaPb_app_opt(j)=ExaAvgBLER(gammadB(j),m,n,app_alpha(:,j),Modulation);
    cmn=((n-m+1)*b(m,n,2)^(n-m+3)+((2*n-2*m+3)*(2*n-2*m+2)/(n-m+2))*m^(n-
m+2))/m/b(m,n,2)^(n-m+2);
    gain_num_opt(j)=10*log10(fsolve(@(g)
Logdiff(g,gammadB(j),m,n,Modulation,exaPb_num_opt(j)),m,options));

```

```

    gain_app_opt(j)=10*log10(fsolve(@(g)
Logdiff(g,gammasB(j),m,n,Modulation,exaPb_app_opt(j)),m,options));
    gain_cf(j)=10*log10(m/(1+cmn/(4*A*gammaW)^(1/(n-m+3)))^(1/(n-m+1))));
end;
end

function out=Logdiff(g,gammasB,m,n,Modulation,targ_Pb)
%fsolve find solution of ExaAvgPb(ones(m,1)*g)=targ_Pb
%'ones(m,1)*g'-uniform distributed power g each branch
%g-the SNR gain
out=log(ExaAvgBLER(gammasB,m,n,ones(m,1)*g,Modulation))-log(targ_Pb);
end

function alpha=App_Alpha(m,n,Modulation)
%Close-form optimum alpha coefficient vector for m by n VELAST system
%alpha-vector of power allocation coefficient
%m-number of Tx antennas
%n-number of Rx antennas
%gammasB-average SNR in dB

%define modulation specifications
gammasB=0:3:39;
if Modulation == 'BPSK'
    A=1; B=1;
else if Modulation == 'QPSK'
    M=4;
    B=2;
    A=sin(pi/M)^2;
else if Modulation=='MPSK'
    M=8;
    B=2;
    A=sin(pi/M)^2;
else if Modulation == 'QAM'
    M=8;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
else if Modulation == 'MQAM'
    M=16;
    B=4*(1-1/sqrt(M));
    A=3/(M-1)/2;
end;
end;
end;
end;
end;
%calculate the alpha(j,:) for large SNR, the power coefficient vector for
%the jth Tx branch
for kk=1:length(gammasB)
    gammaW=10^(gammasB(kk)/10);
    for j=2:m
        alpha(j,kk)=b(m,n,j)*(4*A*gammaW)^(-(j-1)/(n-m+j+1))*(1-
b(m,n,2)/(m*c(m,n,1)*(4*A*gammaW)^(1/(n-m+3))))^c(m,n,j);
    end;
    alpha(1,kk)=m-sum(alpha(2:m,kk));
%normalize alpha(j,kk)
    alpha(:,kk)=m*alpha(:,kk)/sum(alpha(:,kk));
end;
end

function c=c(m,n,j)
%return optimum parameter c(j) for the jth step
c=prod(n-m+1:n+1)/(n-m+j+1);
end

```

```

function b=b(m,n,j)
%return optimum parameter b(j) for the jth step
b=((n-m+j)*nchoosek(2*(n-m+j)-1,n-m+j)*m^(n-m+2)/(n-m+1)/nchoosek(2*(n-m+1)-1,n-
m+1))^(1/(n-m+j+1));
end

```

Appendix C2. Monte Carlo (MC) Simulation

P_e of 8QAM/16QAM with MCR Combining

```

function Pet=MCMRC_QAM(n,Modulation)
warning('off');
gammadB=0:3:39;
Pet = zeros(length(gammadB),1);
for kk = 1:length(gammadB)
    Pe=zeros(MCnumber(gammadB(kk)),1);
    gammaW = round(10^(gammadB(kk)/10));
    %N-symbol transmission numbers in different SNR cases
    N=1e2;NN=5;
    %M-number of channel realizations
    for MM=1:NN
        for M = 1:MCnumber(gammadB(kk));
            %H-channel matrix, n by 1, full rank
            H = 1/sqrt(2)*(randn(n,1) + i*randn(n,1));
            %s-signal symbol stream, 1 by N
            %noramp-normalization coefficient, guarantees unit symbol power
            if Modulation == '8QAM'
                noramp=1/(1/2*(sqrt(2)+(1+sqrt(3))));
                spac=noramp*[1+i,1-i,-1+i,-1-i,1+sqrt(3),-1-sqrt(3),(1+sqrt(3))*i,(-1-
sqrt(3))*i];
            else if Modulation == '16QAM'
                spac=[1+i,1-i,-1+i,-1-i,3+i,3-i,-3+i,-3-i,1+3i,1-3i,-1+3i,-1-
3i,3+3i,3-3i,-3+3i,-3-3i];
                s=randsrc(1,N,[1+i,1-i,-1+i,-1-i,3+i,3-i,-3+i,-3-i,1+3i,1-3i,-
1+3i,-1-3i,3+3i,3-3i,-3+3i,-3-3i]);
            end;
            end;
            xi = 1/(sqrt(gammaW)*sqrt(2))*(randn(n,N) + i*randn(n,N));
            sr = H*s+xi;
            est0=H'*sr/(H'*H);
            est1(1,:)=est0-spac(1);
            est1(2,:)=est0-spac(2);
            est1(3,:)=est0-spac(3);
            est1(4,:)=est0-spac(4);
            est1(5,:)=est0-spac(5);
            est1(6,:)=est0-spac(6);
            est1(7,:)=est0-spac(7);
            est1(8,:)=est0-spac(8);
            z=abs(est1);
            [z1,I]=min(z);
            for k=1:N
                est(k)=spac(I(k));
            end;
            error = sign(abs(est-s(1,:)));
            Pe(M) = sum(error)/N;
        end;
        P(MM)=sum(Pe)/NN;
    end;
end;

```

```

    Pet(kk)= sum(P)/MCnumber(gammapB(kk));
end;

```

BLER of ZF-VBLAST

```

function [BLER,TBER]=VBLAST(m,n,alpha,Modulation)
%Output Avg.BLER and Avg.TBER of optimum power allocation VBLAST
%m-number of Tx antennas
%n-number of Rx antennas
%alpha-optimal power allocation vector (1 by m), sum(alpha)=m
%Modulation-modulation scheme. Only 'BPSK','QPSK' and 'SQAM' in this code

gammapB=0:3:39;
BLER = zeros(length(gammapB),1);
TBER = zeros(length(gammapB),1);
for kk = 1:length(gammapB)
    gammaW = round(10^(gammapB(kk)/10));
    NN=5e1;
    Pb = zeros(NN,1);
    Pet = zeros(NN,1);
    for k=1:NN
        %N-symbol transmission numbers in different SNR cases per iteration
        N=1e2;
        %M-number of channel realizations
        for M = 1:MCnumber(gammapB(kk))
            %H-channel matrix, n by m, full rank
            H = 1/sqrt(2)*(randn(n,m) + i*randn(n,m));
            %s-signal symbol stream, n by N
            %noramp-normalization coefficient, guarantee unit symbol power
            if Modulation=='BPSK'
                noramp=1/1;
                s =noramp*(-1).^round(rand(m,N));
            else if Modulation=='QPSK'
                noramp=1/sqrt(2);
                s=noramp*((-1).^round(rand(m,N))+i*(-1).^round(rand(m,N)));
            else if Modulation=='SQAM'
                noramp=1/(1/2*(sqrt(2)+(1+sqrt(3))));
                spac=noramp*[1+i,1-i,-1+i,-1-i,1+sqrt(3),-1-
sqrt(3),(1+sqrt(3))*i,(-1-sqrt(3))*i];
                s= randsrc(m,N,spac);
            end;
        end;
    end;
    %un-uniform power allocation
    %st-transmitted signal
    for j=1:m
        st(j,:)=sqrt(alpha(j,kk))*s(j,:);
    end;

    %xi-AWGN matrix, n by N
    %sr-recieved signal symbol stream
    xi = 1/(sqrt(gammaW)*sqrt(2))*(randn(n,N) + i*randn(n,N));
    sr = H*st+xi;
    %w-weighting factor in the MPC(Max.SNR) ZF-VBLST

    w = zeros(n,m);
    %Hj-channel matrix after interference nulling in each jth step
    Hj = H;
    for j = 1:m
        Hj(:,1) = [];
    %Pj-projection matrix,Pj=I-Hj*pinv(Hj'*Hj)*Hj'

```

```

%Another way generating Pj is
%Oj = null(Hj');
%Pj = Oj*Oj';

Pj=eye(m)-Hj*pinv(Hj'*Hj)*Hj';

%MMSE weighting
w(:,j) = Pj*H(:,j);
%normalizing w, compensating amplitude variation introduced by H
w(:,j)=w(:,j)/norm(w(:,j)'*H(:,j));
end;

%eb-symbol error counter
%est-detected symbol vector (1 by N) in the jth branch
%est1-detected signal symbol normalized by alpha
y = sr;
eb = 0;
Pe=zeros(m,1);
for j = 1:m
    if Modulation == 'BPSK'
        est0=w(:,j)'*y;
        est1=est0/sqrt(alpha(j,kk));
        est = noramp*sign(real(est1));
    else if Modulation == 'QPSK'
        est0=w(:,j)'*y;
        est1=est0/sqrt(alpha(j,kk));
        est = noramp*(sign(real(est1))+i*sign(imag(est1)));
    %comparing the distances from y to the constellations for MQAM symbol estimation
    else if Modulation == 'MQAM'
        est0 =w(:,j)'*y;
        est1=est0/sqrt(alpha(j,kk));
        est2(1,:)=est1-spac(1);
        est2(2,:)=est1-spac(2);
        est2(3,:)=est1-spac(3);
        est2(4,:)=est1-spac(4);
        est2(5,:)=est1-spac(5);
        est2(6,:)=est1-spac(6);
        est2(7,:)=est1-spac(7);
        est2(8,:)=est1-spac(8);
        z=abs(est2);
    %I-index of the minimum distance
        [z1,I]=min(z);
        for jj=1:N
            est(jj)=spac(I(jj));
        end;
    end;
end;

%interference cancelling from detected symbols in jth step
y=y-H(:,j)*sqrt(alpha(j,kk))*est;
%error-error indicator vector (1 by N) in each signal branch
error = sign(abs(est-s(j,:)));
%eb-total symbol error vector (1 by N) in all m branches
eb = eb + error;
%Pe(j)-avg. symbol error rate in the jth ranch
Pe(j) = sum(error)/N;
end;

%Pb(kk)-Avg. BLER for each avg. SNR setting
%Pet(kk)-total BER for each avg. SNR setting
%sign(eb)-indicating block error in each m parallel transmitting symbols
Pb(k)= Pb(k)+sum(sign(eb))/N; % BLER
Pet(k) = Pet(k)+sum(Pe(:,1))/m; % total BER
end;

%average over MC iterations
Pb(k)= Pb(k)/MCnumber(gammasB(kk));

```

```

        Pet(k) = Pet(k)/MCnumber(gammadB(kk));
    end;
    BLER(kk)=sum(Pb(:))/NN;
    TBER(kk)=sum(Pet(:))/NN;
end;

```

MC Simulation Numbers

```

function N=MCnumber(SNR)
%Estimate MC trials number in one channel realization
%with different channel average SNR input

if 0<=SNR<12
    N=1e4;
else if 12<=SNR<20
    N=1e5;
    else if 20<=SNR<27
        N=1e6;
        else if 27<=SNR<35
            N=1e7;
            else if 35<=SNR
                N=2*1e7;
            end;
        end;
    end;
end;
end

```

Appendix C3. Parallel Monte Carlo (MC) Simulation

Parallel MC Simulation for BLER of ZF-VBLAST

```

pMatlab_Init; %Initialize pMatlab.
Ncpus=pMATLAB.comm_size; %Number of cpus
map1=map([Ncpus 1], {}, 0:Ncpus-1); %Create map for distribution ref.
Pb=zeros(Ncpus,14,map1); %Initial output matrix
Pet=zeros(Ncpus,14,map1);
%Generate numerical power allocation coefficient
[alphaz,alpha]=App_Alpha(2,2,'SQAM');
for i=1:14
    if alpha(1,i)<1
        alpha(:,i)=[1,1];
    end;
end;
%Define the local portion of the distributed matrix
my_Pb=local(Pb);
my_Pet=local(Pet);
%Local simulation running with the alpha coefficient input
[my_Pb,my_Pet]=VBLAST(2,2,alpha,'SQAM');
Pb=put_local(Pb,my_Pb); %Assemble local portion
Pet=put_local(Pet,my_Pet);
%Aggregate distributed matrix to a regular matrix
%with the dimension of (Ncpus,1)
Pb_final=agg(Pb);
Pet_final=agg(Pet);
avg_Pb=sum(Pb_final)/Ncpus; %Average Local results
avg_Pet=sum(Pet_final)/Ncpus;

```