Optimum Ordering for Coded V-BLAST

Alain Uriarte Toboso

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 and Computer Engineering.

University of Ottawa
2012

Abstract

The optimum ordering strategies for the coded V-BLAST system with capacity achieving temporal codes on each stream are studied in this thesis. Mathematical representations of the optimum detection ordering strategies for the coded V-BLAST under instantaneous rate allocation (IRA), uniform power/rate allocation (URA), instantaneous power allocation (IPA) and instantaneous power/rate allocation (IPRA) are derived. For two transmit antennas, it is shown that the optimum detection strategies are based on the per-stream before-processing channel gains. Based on approximations of the per-stream capacity equation, closed-form expressions of the optimal ordering strategy under the IRA at low and high signal to noise ratio (SNR) are derived. Necessary optimality conditions under the IRA are given. Thresholds for the low, intermediate and high SNR regimes in the 2-Tx-antenna system under the IPRA are determined, and the SNR gain of the ordering is studied for each regime. Performances of simple suboptimal ordering strategies are analysed, some of which perform very close to the optimum one.

Table of Contents

Table of Figures

List of Acronyms

Acronym	Meaning
APA	average power allocation
APRA	average power/rate allocation
ARA	average rate allocation
AWGN	additive white Gaussian noise
BER	bit error rate
BLER	block error rate
BPSK	binary phase-shift keying
CDF	cumulative distribution function
CSI	channel state information
D-BLAST	Diagonal Bell Labs Layered Space-Time
FWF	fractional waterfilling
GSO	Gram-Schmitt Orthogonalization
IPA	instantaneous power allocation
IPRA	instantaneous power/rate allocation
IRA	instantaneous rate allocation
ISTI	inter-stream interference
MAC	multiple access channel
MIMO	multiple-input multiple output
MISO	multiple-input single-output
ML	maximum likelihood
MMSE	minimum mean-square error
M-PSK	M-ary phase-shift keying
M-QAM	M-ary quadrature amplitude modulation
MRC	maximum ratio combining
NC EGC	noncoherent equal gain combining
QoS	quality of service
Rx	receiver
SER	symbol error rate
SIC	successive interference cancellation
SIMO	single-input multiple-output
SISO	single-input single-output
SNIR	signal to noise plus interference ratio
SNR	signal to noise ratio
SVD	singular value decomposition
TBER	total bit error rate
Tx	transmitter

Acronym	**Meaning**
URA	uniform power and rate allocation
V-BLAST	Vertical Bell Labs Layered Space-Time
WF	waterfilling
ZF	zero-forcing

List of Symbols and Notations

Notation	Meaning
T	transposition
$+$	conjugate transposition
$\perp$	orthogonal projection
$\lvert\mathbf{v}\rvert$	Euclidean norm of vector **v**
$\mathbf{V}$	matrix **V**
v_i	i-th component of vector $\mathbf{v} = [v_1, v_2, ..., v_m]^T$
$\mathbf{v}_i$	i-th column of matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m]$
$\mathbf{v}_{i\perp}$	projection of the i-th column of matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m]$ orthogonal to the subspace spanned by the $m-i$ right-side columns

Symbol	Meaning	First appearance
m	number of transmit antennas	Section 2.1
n	number of receive antennas	Section 2.1
$\mathbf{q} = [q_1, q_2, ..., q_m]^T$	Tx vector	(3.1)
$\mathbf{r} = [r_1, r_2, ..., r_n]^T$	Rx vector	(3.1)
$\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_m]$	MIMO channel matrix	(3.1)
$\xi \sim CN(0, \sigma_0^2 \mathbf{I})$	circularly symmetric additive white Gaussian noise vector with i.i.d. entries	(3.1)
σ_0^2	noise power	(3.1)
$\mathbf{P}_i$	projection matrix orthogonal to the spatial signatures of the $m-i$ yet to be detected symbols	Section 3.5
$\bar{\xi} \sim CN(0, \mathbf{P}_i \sigma_0^2)$	correlated noise vector after the interference nulling step	(3.4)
$\mathbf{w}_i$	optimum ZF weights	(3.5)
π^*	optimum ordering	(3.8)
$\xi_i^{'} \sim CN(0, \sigma^2)$	scalar noise after applying the optimum ZF weights	(4.1)
$\gamma_0 = 1/\sigma_0^2$	average SNR at each receive antenna	(4.2)
P_{out}	outage probability	(4.4)
C_{IRA}	capacity under the IRA	(4.7)
$C(\pi_i)$	system capacity under ordering π_i	(4.7)

Symbol	Meaning	First appearance
C_{IRA}	capacity under the IRA	(4.7)
$\mathbf{h}_{i\perp j}$	projection of $\mathbf{h}_i$ orthogonal to $\mathbf{h}_j$ where φ is the angle between both vectors	(4.7)
N	maximum number of orderings that can satisfy the necessary optimality conditions	(4.27)
C_{URA}	capacity under the URA	(5.1)
$\mathbf{\Lambda} = diag\left(\sqrt{\alpha_1},...,\sqrt{\alpha_m}\right)$	diagonal matrix which entries represent the squared root of the power assigned to each stream	(6.1)
C_{IPA}	capacity under the IPA	(6.6)
$\bar{g}(\pi) = \left(\frac{1}{m}\sum_i \left(g_{i\perp}(\pi)\right)^{-1}\right)^{-1}$	harmonic mean per-stream power gain for a given ordering π	(6.7)
C_{IPRA}	capacity under the IPRA	(6.27)
C_{WF}	capacity under the waterfilling	(6.27)
C_{FWF}	capacity under the fractional waterfilling	
$\mu(\pi)$	water level (from the WF algorithm) for a given ordering π	(6.29)
$\alpha_i^*(\pi) = \left[\mu(\pi) - \frac{1}{\gamma_0 \left\|\mathbf{h}_{i\perp}(\pi)\right\|^2}\right]_+$	optimum power allocation provided by the WF algorithm for a given ordering π	(6.29)
G	SNR gain of the optimum ordering procedure	(7.1)

Acknowledgments

This thesis would not have been possible without the support, guidance and patient of my direct supervisor Dr. S. Loyka and co-supervisor Dr. Francois Gagnon. I would also like to thank my brother J. A. Uriarte for his special help, advices and for always being there no matter time or day of the week.

My gratitude also goes to my family for running always by my side despite being separated by thousands of kilometers. Special thanks to my dad for being the winner of many battles, you are an inspiration for me. My deep thankfulness to my mom, brother, wife and grandmother for their unconditional support and love through all this process.

1. Introduction

1.1. Motivation

The Multiple-input multiple-output (MIMO) communication architecture has been widely studied during the last 15 years due to the fact that it provides very high spectral efficiencies that cannot be attained by conventional techniques [1]-[4]. However, this increase in the spectral efficiencies is accompanied by a significant growth in the system complexity.

Vertical Bell Labs Layered Space-Time (V-BLAST) was proposed by Foschini [3] as a low complexity MIMO scheme able to achieve a substantial portion of the total MIMO capacity given that the multipath environment is rich enough. In the V-BLAST architecture, multiple data streams are transmitted over the multiple transmit (Tx) antennas simultaneously, which are detected at the receiver using successive interference cancelation to achieve good system performance at moderate complexity. The order at which the streams are detected affects the V-BLAST performance.

Unordered V-BLAST has been commonly used to study the performance of this architecture during the last years; optimization strategies that help enhance its performance have been proposed in [24], [31], and [32]. Meanwhile, ordered V-BLAST represents a challenge to analytical examination due to the increased complexity added by the ordering procedure.

The optimum ordering procedure for the uncoded V-BLAST in Rayleigh fading channels was proposed by Foschini in [3]. The stream detection order is organized according to their after processing SNRs in the decreasing order, i.e. at each step the remaining stream with highest after processing SNR is detected first and then its contribution is subtracted from the received vector for next detection steps. The optimality of this ordering strategy is based on the fact that it minimizes the total error probability of the system. On the other hand, a closed-form analysis of the optimum detection ordering for the coded V-BLAST has not been settled yet.

A closed-form analysis of the optimum detection ordering for the coded V-BLAST is provided in this thesis. The optimum orderings under the IRA, the URA, the IPA and the IPRA are studied. Any optimization strategy in coded systems must target the outage probability. Since the instantaneous optimizations of the outage probability and the capacity achieve the same lowest value of the outage probability in the coded V-BLAST [32], the optimization of the detection ordering is studied from the system capacity point of view.

1.2. Contributions of the thesis

The main contributions of this thesis are as follows:

- Derivation of the optimal ordering strategies in the coded V-BLAST under the IRA, the URA, the IPA and the IPRA.
- Comprehensive closed-form analysis of the optimal ordering strategies under the IRA, the URA, the IPA and the IPRA when using two Tx antennas.
- Closed-form expressions of the optimal ordering in the coded V-BLAST under the IRA at low and high SNR based on approximations of the per-stream capacity equation.
- Derivation of necessary optimality conditions for an ordering strategy in the coded V-BLAST under the IRA.

- Derivation of SNR thresholds that separate the low, the intermediate and the high SNR regimes in the coded V-BLAST with two transmit antennas under the IPRA.
- Definition and closed-form analysis of the SNR gain of ordering in the coded V-BLAST with two transmit antennas under the IPRA.

1.3. Thesis outline

The main goal of this thesis is the closed-form analysis of the optimum stream detection ordering in the coded zero-forcing V-BAST. Chapter 2 gives a review of the relevant research carried out in the wireless MIMO field, devoting special attention to the BLAST architecture. Chapter 3 introduces the channel model used along the thesis and describes the V-BLAST algorithm. Chapters 4 and 5 provide the closed-form analysis of the optimal detection ordering under the IRA and the URA respectively. The optimal detection orderings under the IPA and the IPRA are investigated in Chapter 6. Chapter 7 defines the SNR gain of ordering and gives a detailed analytical breakdown to the SNR gain of ordering when using two Tx antennas under the IPRA. Chapter 8 extends the results obtained from the point-to-point perspective to the multiuser communications viewpoint. Finally, Chapter 9 states the conclusion down from the results presented in the thesis and outlines areas for future research.

2. Literature review

2.1. MIMO systems and channel capacity

Designing wireless communications systems with high spectral efficiencies and high quality of service (QoS) represents a significant engineering challenge. Several strategies have been used to increase the data rate that can be sent through a channel with arbitrary small error probability, i.e. the channel capacity. One approach consists of increasing the bandwidth so that more bits can be transmitted to the medium per unit time (increasing the data rate in bits/sec). However, bandwidth is a very limited and expensive resource and in addition, increasing the bandwidth does not increase the spectral efficiency (in bits/sec/Hz) of the system. Another alternative is to increase the transmit power (P_t) since spectral efficiency is an increasing function of P_t, nevertheless most communication systems are power limited due to interference and/or human health concerns.

Due to the multipath characteristic of the wireless propagation channel, multiple copies of the transmitted signal arrive at the receiver at different moments of time. This combination of signals with a phase difference at the receiver causes abrupt variations in the received signal power (P_r) or in the received SNR, which is known as fading. Fading creates outage events and further limits the capacity of the wireless channel; however it can be mitigated using diversity techniques.

Diversity techniques use independent channels to send different replicas of the desired signal to the receiver [33]. Diversity techniques can operate in time, frequency and/or space domains. The two representations of space diversity techniques are transmit diversity where multiple transmit antennas are used (also known as multiple-input single-output or MISO systems) and receive (Rx) diversity where multiple receive antennas are used (also known as single-input multiple-output or SIMO systems). In both cases it is required for antennas to be placed sufficiently far apart so that the channel gains between different antenna pairs fade independently. These systems provide a diversity gain that is reflected in the exponent of SNR in the error probability equation. If the error probability of a system is expressed as

$$P_e \approx \frac{a}{\gamma_0^L} \tag{2.1}$$

where γ_0 is the average SNR, then the diversity gain or diversity order of the system is L and a is the SNR gain. The error probability decreases as the L-th power of SNR, corresponding to a slope of $-L$ in the error probability curve (in dB scale).

MISO and SIMO systems offer higher capacity than single-input single-output (SISO) systems without increasing power or bandwidth; however the capacity increases logarithmically with the number of transmit or receive antennas respectively [1].

In 1995-1996, the multiple-input multiple-output (MIMO) wireless system architecture was proposed by Foschini [2] and Telatar [1] as a spectrally efficient way of communication. The key idea of MIMO systems is to transmit multiple data streams using a set of Tx antennas and to use multiple Rx antennas and appropriate signal processing to recover them. It was shown in [2] that with MIMO the capacity of the wireless (uncorrelated) propagation channel increases linearly with $\min(m,n)$ where m and n are the number of Tx and Rx antennas respectively. Experimental results in [3] showed that the MIMO capacity was more than 10 times higher than that of the SISO system for the same bandwidth and total transmit power constraint. These pioneering works generated a great interest in the area of MIMO systems around the word. The number of publications in this area has been enormous since its discovery in 1995-1996 (Figure 1 shows the details).

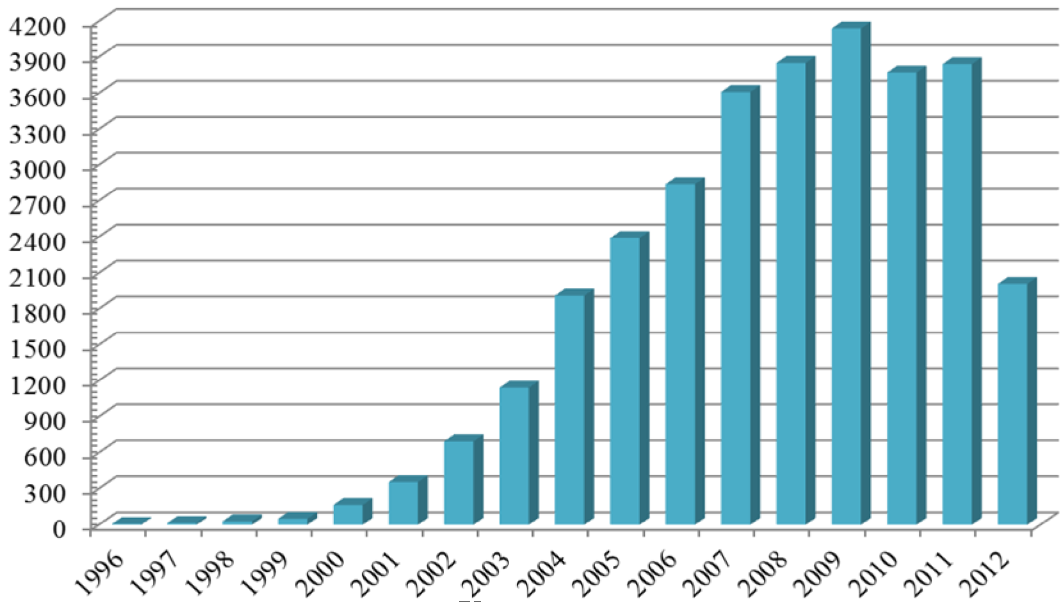

Figure 1: Number of MIMO publications since its discovery.[1]

2.2. BLAST architecture

The BLAST architecture, introduced by Foschini [2] in 1996, is a low complexity transceiver architecture to communicate over the MIMO wireless channel. Although suboptimal, it is able to attain a significant fraction of the theoretical MIMO capacity over the rich-scattering wireless channel. BLAST implies independent transmission of streams at the transmitter side and successive interference cancellation (SIC) at the receiver.

In the BLAST architecture (see Figure 2), independent data streams are transmitted at the same time and frequency using a set of Tx antennas. At the receiver end, the detection of a given stream includes three main procedures:

[1] This data includes all published paper containing all the keywords “MIMO”, “wireless”, “channel”, “space-time”, “communications” returned by the Google Scholar search engine, for each year.

- interference cancellation from already detected streams,
- interference nulling of yet-to-be-detected streams,
- optimal ordering

Under this scenario, each data stream is decoded independently after nulling the interference generated by the yet-to-be-detected streams and after canceling the interference from the already detected ones. During the interference cancellation procedure the stream is re-encoded and its contribution is subtracted from the Rx vector. The order at which the streams are detected affects the general performance of the BLAST architecture and hence an ordering procedure is necessary as well.

The original detection algorithm for the BLAST architecture is the combination of linear nulling and SIC. The nulling vectors can be generated by the zero-forcing (ZF) or the minimum mean-square error (MMSE) criterion, thus the corresponding algorithm is generally called ZF-SIC or MMSE-SIC algorithm.

In the original transmission process for the BLAST architecture proposed in [2], instead of assigning each of the independent streams to a specific antenna, the bitstream/antenna association is periodically cycled, i.e. each stream is dispersed "diagonally" across antennas and time. The BLAST architecture under this transmission strategy is known as Diagonal BLAST (D-BLAST). The complexity of the D-BLAST may be too high for practical systems.

In 1998, V-BLAST (Vertical BLAST) was introduced in [3] as a low complexity wireless communication architecture. In the V-BLAST, the layering is horizontal, meaning that all the symbols of a certain stream are transmitted through the same antenna (one stream per antenna). It was shown in [4] that this architecture is able to achieve very high spectral efficiencies e.g. spectral efficiencies in the order of 20 – 40 bits/sec/Hz in an indoor propagation environment at realistic SNR's and error rates. More details of the V-BLAST algorithm will be discussed in the next chapter.

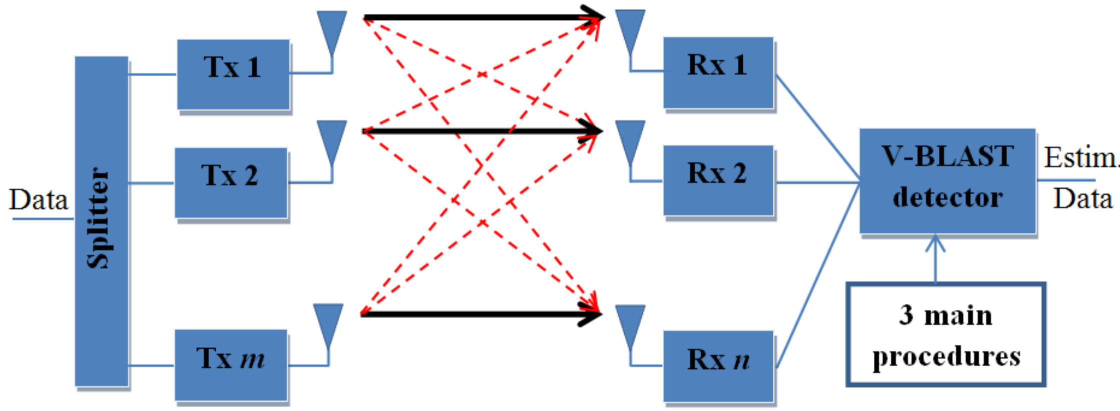

Figure 2: BLAST architecture.

V-BLAST has interested researchers from all over the word due to its enormous potential. Many papers have been published addressing different issues since its discovery. Significant research effort has been made not only to reduce V-BLAST complexity but also to improve its performance. The V-BLAST performance under real conditions environments, where channel estimations errors occur, has also been a field of wide research.

2.3. Reducing complexity of the V-BLAST

The complexity of the V-BLAST algorithm lies on two principal factors: the detection strategy used and the optimal ordering procedure. Addressing these concerns Wai et al. proposed in [5] the replacement of the optimal decoding order by a suboptimal one, based on the pseudo-inverse of the channel matrix, and the utilization of Gram-Schmitt Orthogonalization (GSO) to compute the pseudo-inverse in finding the weight vectors in the original V-BLAST. They obtained a 71% reduction of the total number of arithmetic operations for a 12x8 system in slow fading channel. A recursive MMSE-SIC algorithm was presented in [7]. The MMSE nulling vectors and the optimal detection order were calculated from the previous computational results via simple recursive pseudo-inverse formulas. The complexity of the proposed algorithm was shown to be lower than that proposed in [6], where another fast recursive algorithm was presented using the Sherman-Morrison formula and the principle of partitioned matrices. The reduction in complexity in terms of multiplications, additions and floating-point operations was more evident for a practical (small) number of transmit antennas.

Efficient detection algorithms utilizing the QR decomposition[2] of the channel matrix were proposed in [9]-[13]. An algorithm that jointly calculates an optimized detection order and the QR decomposition of the channel matrix was proposed in [10]-[11] (MMSE Sorted QR Decomposition). Hassibi [9] proposed a “square-root” MIMO detection algorithm based on detecting first the symbol associated with the maximum diagonal entry in the R matrix after the QR decomposition of the channel matrix. This algorithm was improved by Zhu et al. [12] by a 36%, reduction in the number of multiplications and additions. A new improvement of the already improved “square-root” algorithm was proposed in [13] based on a fast algorithm for inverse Cholesky factorization. These detection algorithms based on QR decomposition reduce the number of matrix inversion and represent an interesting alternative.

[2] QR decomposition is the decomposition of a matrix **H** into a product **H**=**QR** of an orthogonal matrix **Q** and an upper triangular matrix **R**.

Based on the standard MMSE V-BLAST algorithm, a reduced complexity detection algorithm (RC-MMSE-SIC) was proposed in [8]. The main idea of the proposed algorithm is to detect the streams whose signal to interference plus noise ratios (SINRs) are above a certain threshold instead of detecting only the stream with largest SINR in each detection step as proposed by the original V-BLAST algorithm. The algorithm also makes use of the GSO to compute the pseudo-inverse in finding the weight vectors. The scheme, although suboptimal, decreases the computational complexity of the standard MMSE V-BLAST.

2.4. Performance analysis and improvement of the V-BLAST

Significant research effort has been made to analyse and improve the V-BLAST's performance. In [14], a geometrically-based analytical approach to the performance analysis of the ZF V-BLAST algorithm was presented. It was shown that without optimal ordering and under uncorrelated Rayleigh fading channel, the diversity order at the *i*-th processing step is $n-m+i$, where n and m represent the number of receive and transmit antennas respectively. Outage probabilities and average bit error rates (BERs) expressions were derived for the specific case of $2\text{x}n$ (two transmit and n receive antennas) systems when optimal ordering is implemented. Moreover it was shown that the effect of the optimal ordering in $2\text{x}n$ systems is a SNR gain of 3 dB at the first detection step and no diversity gain is attained. However, the use of noncoherent equal gain combining (NC EGC) after the interference nulling (orthogonal projection) was assumed in the analysis, which is not optimum, and the after-projection noise correlation was ignored. Furthermore, the error propagation was disregarded.

In [16], an analytical performance evaluation of the unordered ZF V-BLAST in Rayleigh fading channels was made, this time the optimum maximum ratio combining (MRC) was employed after the interference projection while taking into account the after projection noise correlation. The MRC weights provide the best performance in terms of the output SNR, and hence, the BER. It was demonstrated that the optimum MRC weights include the projections and are orthogonal to each other resulting in the after-combining noise components to be independent at each step; then closed-form expressions for the instantaneous BER at each step were derived. Average BER expressions were also obtained based on the facts that the instantaneous SNR at each step are independent of each other and that the inter-stream interference (ISTI) is Gaussian for a Rayleigh independent and identically distributed (i.i.d.) channel. Exact BER expressions, taking into account the error propagation, were obtained for binary phase-shift keying (BPSK) showing that while the error propagation affects dramatically higher detection steps (resulting in the diversity order being equal to $n-m+1$ at each step), the first detection step is not affected. As the first stage dominates the error performance (it has the lowest diversity order when error propagation is ignored), it was concluded that the error propagation has only a minor effect on the total average BER. While the assumption of no ordering allowed making such an insightful evaluation, it limited the results obtained.

In [18], an analytical approach to the analysis of the $2\text{x}n$ V-BLAST was presented and the results were shown to be consistent with those in [14]. In [19], previous work in [14] was extended, the authors evaluated the outage and error rate performance of the ordered ZF V-BLAST with more than two transmit antennas in i.i.d Rayleigh fading channels using a geometrically-based framework. Based on a number of bounds on the outage probability, accurate closed-form approximations to the average block error rate (BLER[3]) and the total bit error rate (TBER[4]) were derived. It was shown that for m transmit antennas, the effect of the optimal ordering is an m-fold SNR gain at the first step, but no diversity gain is obtained. This work also extended the analysis made in [15] and [17] where from a diversity-order-based analysis it was demonstrated that the diversity order is not affected by the ordering procedure. The rigorous mathematical proof of the m-fold SNR gain of ordering was derived in [25].

The analytical approaches to V-BLAST provide significant insight into the algorithm performance and its bottlenecks creating the base for optimization. In the V-BLAST algorithm, lower detection steps enjoy lower diversity order (ignoring the error propagation) limiting the system performance. In this sense, a widely used technique for improving the error performance of the uncoded V-BLAST is to use a non-uniform power allocation that reduces the error rates at early stages.

Either instantaneous/average BLER or instantaneous/average TBER have been used as the objective functions to minimize in the problem of finding the optimum power allocation. In [22], the optimum transmit power allocation was numerically obtained for $2\text{x}n$ V-BLAST using the instantaneous BLER as the objective function. An approximation of the transmit power vector that minimizes the instantaneous TBER (ignoring the error propagation) was derived in [21], [23]. Meanwhile in [20], closed-form expressions for the optimum power allocations of the uncoded ZF V-BLAST and MMSE V-BLAST, using the instantaneous TBER and accounting for error propagation, were derived based on a number

[3] It is defined in [19] as the probability of having at least one error in the detected transmit symbol vector.

[4] It is defined in [19] as the error rate of the output stream to which all the individual sub-streams are merged after the detection.

of approximations. It was also shown in [20] (via simulations), that the error propagation does not have a significant impact in the performance of the optimized systems. In [24], compact closed-form approximations for the optimum power allocations, based on the average BLER and TBER were obtained. It was demonstrated that the SNR gain of the optimum power allocation cannot exceed the number of transmit antennas and that the optimization based on the TBER results in the same performance as the one based on the total BLER. The latter was shown to be more suitable for analytical techniques since it does not require explicit characterization of the error propagation effect.

2.5. V-BLAST under channel estimation errors

Most V-BLAST detection algorithms i.e. ZF-SIC or MMSE-SIC and optimization techniques are based on perfect channel knowledge being available at the receiver. However, perfect channel knowledge is never available a priori. In practice, the channel has to be estimated. This can be done, for example, by transmitting pilot symbols that are known in advance at the receiver. As the system performance depends on the quality of the channel estimate, extensive research has been carried out to study the impact of channel estimations errors on V-BLAST architecture.

In [26] the performance of the uncoded V-BLAST under channel estimation errors was analyzed. The performance was examined through a perturbation analysis. The perturbation of the channel matrix was approximated by an additional noise term added to the original unperturbed system suggesting that under channel estimation error the V-BLAST system will suffer from additional system noise. A tight error floor was derived as a result of the equivalent system noise, which is a combination of the channel estimation errors and the additive white Gaussian noise at the receiver. It was shown via simulations that the MMSE is not more robust than the ZF receiver under channel estimation errors. Simulations also showed that ZF-SIC is a more robust option able to tolerate about twice the amount of channel estimation errors as the ZF receiver. However, the authors did not consider the effect of channel estimation error in the ZF receiver caused by computing the pseudoinverse of the inaccurate channel estimate. Furthermore, they focused on the approximation of the before-processing SNR.

An error-propagation analysis of the uncoded ZF V-BLAST with channel estimation errors was carried out in [27]. A tight upper bound for the average symbol error rate (SER) was derived. Furthermore, it was pointed out that the V-BLAST processing with channel-estimation errors and the detection order based on perfect channel estimates produce no significant change in the system performance (in terms of the average SER).

The effect of channel estimation errors on the performance of MIMO ZF receivers in uncorrelated Rayleigh flat fading channels was investigated in [28]. This time the focus was on the approximation of the after-processing SNR distribution. By modeling the estimation error as independent complex Gaussian random variables, tight approximations for both the after-processing SNR distribution and bit error rate (BER) for MIMO ZF receivers with M-QAM and M-PSK modulations were derived in closed-form. Besides the previously mentioned error floor, it was found that the BER under channel estimations error is an increasing function of the number of Tx antennas.

An analytical method to derive the average SER of the signals detected at each step in ZF V-BLAST was presented in [30]. The method accounts for both error propagation and channel estimation errors. It was shown that ZF V-BLAST is more sensitive to channel estimation errors at high SNR, and that the effects of inaccurate channel estimations have more significant impact in later detection steps when optimal order is employed. Nevertheless, at high SNR channel estimation errors are less probable to occur.

Some researchers have proposed modifications to the original V-BLAST algorithm in order to improve its performance under channel estimation errors. For example a robust symbol detection ordering method for the uncoded ZF V-BLAST was proposed in [29] (robust in the sense to be less sensitive to channel estimation errors). The proposed ordering was shown to optimize the average post-detection SINR over the channel estimation errors and to organize the detection order by decoding the symbol corresponding to the best average SINR first. Although the authors claimed that the proposed algorithm is capable of achieving global performance optimization, this scheme does not take into account the error propagation due to the SIC.

The error propagation is present in the uncoded V-BLAST and hence it limits the overall system performance. However, in the coded V-BLAST the use of capacity achieving temporal codes at each stream allows the per-stream transmission rates to match the corresponding capacities, so there are no errors when the streams are not in outage, and hence no error propagation in-between the streams. This thesis is mainly devoted to the study of the optimum ordering strategies in the coded ZF V-BLAST. Given our focus, the use of capacity achieving temporal codes at each stream and perfect channel estimation at the receiver will be assumed.

2.6. The coded V-BLAST

While the studies mentioned above deal with the uncoded V-BLAST, uncoded systems are rare and most modern communication systems use coding [33]. Average and instantaneous optimization of power and rate allocation for the coded V-BLAST have been studied in [31] and [32]. The performance metrics in these systems are the outage probability and the outage capacity. In [31], an analysis and performance evaluation of three average optimization strategies targeting the outage probability under the total power constraint was carried out. The three optimization strategies were: average power allocation (APA), motivated by the fact that many practical system use power control; average rate allocation (ARA), suitable for variable-rate system using identical and fixed power amplifiers to simplify the RF part of the system; and jointly average power and rate allocation (APRA), which is suitable for variable-rate variable-power systems. It was shown that the APA offers an SNR gain (upper-bounded by the number of transmit antennas), but the diversity order of the system remains unchanged. This is the same result obtained for the uncoded V-BLAST in [24]. The ARA increases the system diversity order (the diversity orders at each stream are equal) and hence is more efficient than the APA. The APRA only offers a power gain over the ARA, but no diversity gain. The same study was made for the case of instantaneous optimization in [32]. In this case the three optimization strategies studied were: instantaneous power allocation (IPA), instantaneous rate allocation (IRA), and jointly instantaneous power and rate allocation (IPRA). It was demonstrated that the maximization of the instantaneous system capacity (via the IPRA) also minimizes the outage probability and, hence, both problems are equivalent under arbitrary fading distribution. It was also proven that the conventional waterfilling (WF) algorithm is not optimal for V-BLAST. Instead the fractional waterfilling (FWF) algorithm was proposed and shown to maximize the V-BLAST capacity via the IPRA. Furthermore it was demonstrated that this algorithm attains the full MIMO channel diversity in the low outage regime. An optimum instantaneous power allocation to maximize the system capacity of a multi-stream transmission under uniform power and rate allocation was also presented.

Either in the presence of uncoded or coded V-BLAST it is a general agreement that the instantaneous optimization offers better performance than the average one, but at the cost of increasing the system complexity due to the necessary feedback and power reallocation for every channel realization.

The results obtained in [31] and [32] are very insightful and will be widely used here. However, they are limited because the optimal ordering was not considered. The aim of this thesis is precisely to help fill this gap by studying optimal ordering strategies in the coded V-BLAST.

2.6. Summary

MIMO is one of the most important technological discoveries in the wireless communication field. MIMO systems offer theoretical transmission rates over the wireless propagation channel never imagined before. However, the high complexity associated with MIMO technology is the main limitation for some applications.

V-BLAST is a transceiver architecture designed to attain a significant portion of the theoretical capacity offered by the MIMO wireless propagation channel at a relatively low complexity. Substantial research efforts have been made to reduce the V-BLAST complexity and to analyse/improve its performance under both ideal and realistic conditions. A literature review outlining the most significant research about this architecture has been provided in this chapter. While several papers dealing with the detection ordering in the uncoded V-BLAST have been published, little is known about the detection ordering when coding is used. In order to shed light on this issue, this thesis studies the optimum detection ordering in the coded V-BLAST when capacity achieving temporal codes are used at each stream.

3. The V-BLAST algorithm

3.1. Transmission strategies for MIMO communications

Modern wireless communications systems demand high data rate accompanied with high reliability i.e. low error probability. MIMO system has shown to offer very high spectral efficiencies in rich scattering environments, its capacity scales lineally with the minimum number of antenna elements [1]-[2]. When the channel matrix is perfectly known at the transmitter, the full capacity of the wireless MIMO channel can be achieved by transmitting independent streams in the directions of the right singular vectors of the channel matrix and assigning the powers following the well-known waterfilling algorithm.

Under the previous approach, perfect channel state information (CSI) must be available at the transmitter. Also a singular value decomposition (SVD) of the channel matrix, which is a very complex and time consuming operation, is required. Furthermore, in the case of fading channels the channel state changes constantly and the SVD of the channel matrix has to be executed for every realization of the channel overwhelming the system.

In practical systems, a simple and efficient approach is to send independent data streams through the different transmit antennas. This transmission strategy does not require the knowledge of the channel matrix at the transmitter side. Nevertheless, by having this information at the transmitter, instantaneous power/rate allocation can be used in order to improve the system performance. This transmission strategy is also able to achieve the wireless MIMO channel capacity when appropriate signal processing is used for the detection of each stream at the receiver.

3.2. Receiver architectures for MIMO communications

Due to the nature of the wireless propagation channel, in MIMO systems a mixture of the signals transmitted from each transmit antenna impinges over each receive antenna. The objective of the receiver is to recover the signals from each transmit antenna in a reliable and computationally effective way.

The capacity of the wireless MIMO channel can be attained by jointly decoding the received data streams using the Maximum Likelihood (ML) receiver. However, the complexity of this method grows exponentially with the number of streams making it unfeasible for systems with a high number of transmit antennas.

The complexity of the ML receiver can be reduced using linear receivers as MMSE and ZF receivers. These receiver architectures use linear operations to convert the problem of joint decoding of the data streams into one of individual decoding of the data streams [33].

The MMSE receiver optimally trades off fighting inter-stream interference and isotropic Gaussian noise. It maximizes the output SINR for any value of SNR. The MMSE receiver can be used for the detection of each stream separately. In the detection of a given stream it can be interpreted as a receiver that first whitens the spatially-colored noise (represented by the sum of the inter-stream interference and the isotropic Gaussian noise) and then applies MRC for the case of white Gaussian noise to maximize the output SNR [33].

The ZF receiver focuses on completely nulling out the inter-stream interference disregarding the presence of noise. This is done through the multiplication by a projection matrix which is orthogonal to the subspace spanned by the spatial signatures of the yet to be detected streams. Then the demodulation of the given stream can be performed match filtering to the projected channel gain vector. This receiver maximizes the output SNR subject to the constraint of nulling out the interference from other streams.

3.3. Successive interference cancellation (SIC)

The performance of the MMSE and ZF receivers can be improved by the successive cancellation of the already detected streams. Once a data stream is successfully recovered, its contribution can be subtracted from the received vector and next detected streams will not face the interference caused by the already detected one.

This combination of MMSE or ZF receivers with SIC is precisely the detection algorithm used for the V-BLAST architecture. The corresponding algorithm is known as ZF-SIC (or ZF V-BLAST) if the ZF receiver is used or as MMSE-SIC (or MMSE V-BLAST) if the MMSE receiver is employed.

The MMSE-SIC is able to achieve the full capacity of the wireless MIMO channel while the ZF-SIC can attain a substantial portion of it at a lower complexity. Due to its lower complexity and treatable equations, the ZF-SIC will be used to evaluate the optimal detection ordering of the coded V-BLAST.

3.4. V-BLAST: Channel model and assumptions

The baseband MIMO channel model employed in the thesis is:

$$\mathbf{r} = \mathbf{H}\mathbf{q} + \boldsymbol{\xi} \qquad (3.1)$$

where $\mathbf{q} = [q_1, q_2, ..., q_m]^T$ and $\mathbf{r} = [r_1, r_2, ..., r_n]^T$ are the transmitted and received signal vectors respectively, $\mathbf{H}$ is the nxm (n Rx and m Tx antennas) channel matrix $(n \geq m)$ with its $(i, j)-$th entry representing the complex channel gain from transmit antenna j to receive antenna i; as Rayleigh fading channel is assumed, $\mathbf{H}$ is modeled with independent, identically distributed (i.i.d.) circularly symmetric standard complex Gaussian entries, denoted as $h_{ij} \sim CN(0,1)$. $\boldsymbol{\xi}$ is the circularly symmetric additive white Gaussian noise vector with i.i.d. entries i.e. $\boldsymbol{\xi} \sim CN(0, \sigma_0^2 \mathbf{I})$.

The column-wise representation of the channel matrix $\mathbf{H}$ is given by $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_m]$ where $\mathbf{h}_i$ is a column vector containing the channel gains from the *i*-th Tx antenna to all Rx antennas. In fact $\mathbf{h}_i$ is a circularly symmetric standard complex Gaussian random vector i.e. $\mathbf{h}_i \sim CN(0, \mathbf{I})$. The system model in (3.1) can also be represented as

$$\mathbf{r} = \sum_{i=1}^{m} \mathbf{h}_i q_i + \boldsymbol{\xi} \tag{3.2}$$

Other assumptions are as follows. A flat fading environment is assumed, where the channel remains constant during a frame of information bits but it may vary from frame to frame i.e. slow block fading channel. The channel can be perfectly tracked by the receiver (no channel estimation error) and, in the cases where feedback is required to the transmitter, the feedback session is executed without errors. ZF-SIC is assumed for signal detection. Capacity-achieving temporal codes are used for each stream in the V-BLAST so that the per-stream rates match the corresponding capacities and there are no errors when streams are not in outage and, therefore, no error propagation in-between the streams. The transmitted signal, noise and channel gains are independent of each other and there is no performance degradation due to synchronization and timing errors.

3.5. The V-BLAST architecture using per-stream coding

In the coded V-BLAST architecture (see Figure 3) the incoming bit stream is demultiplexed into m data streams. These streams are then encoded using capacity achieving temporal Gaussian codes and transmitted in parallel at the same time and frequency using a set of m Tx antennas. At each receive antenna the signals interfere with each other due to the effect of the wireless propagation channel. By using ZF-SIC, an efficient signal processing procedure implemented at the receiver side, the interference (caused by the other streams) at each receive antenna is eliminated. This procedure transforms the wireless propagation channel into a set of virtually independent sub-channels [16]. The ZF V-BLAST algorithm has three main steps: (1) interference cancellation, (2) interference nulling and (3) optimal ordering.

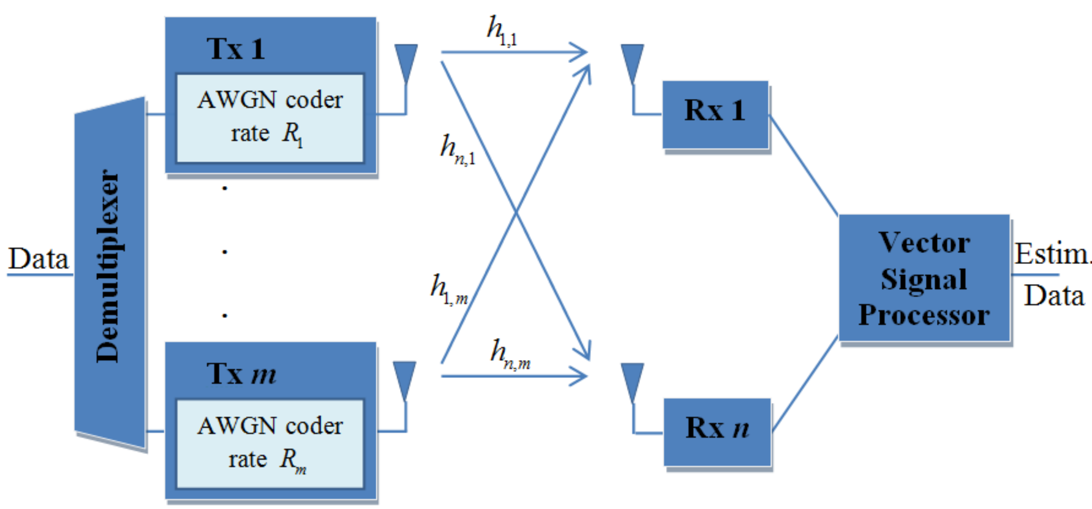

Figure 3: Pictorial representation of the V-BLAST architecture when coding is used at each stream.

For a better understanding of the algorithm, the interference cancellation and nulling steps are discussed first for a given ordering, and then the system model is extended to the case where an optimal ordering is employed.

Interference cancellation: At the *i*-th step (when the *i*-th Tx symbol is detected), the interference from the $(i-1)$ already received symbols can be subtracted from the received vector based on their estimations $(\hat{q}_1, \hat{q}_1, ..., \hat{q}_{i-1})$ and on the knowledge of the channel matrix at the receiver end:

$$\mathbf{r}_i ' = \mathbf{r} - \sum_{j=1}^{i-1} \mathbf{h}_j \hat{q}_j \qquad (3.3)$$

Interference nulling: After the interference cancellation step, the interference from yet to be detected symbols can be nulled out projecting the received vector at this step orthogonal to the subspace spanned by the yet to be detected symbols [16]. This is accomplished by multiplying the received vector by a matrix $\mathbf{P}_i$ orthogonal to the spatial signatures of the $m-i$ yet to be detected symbols[5] i.e. $\mathbf{P}_i \perp \mathbf{H}_i = [\mathbf{h}_{i+1} ... \mathbf{h}_m]$. Figure 4 illustrates the geometrical representation of the interference nulling step.

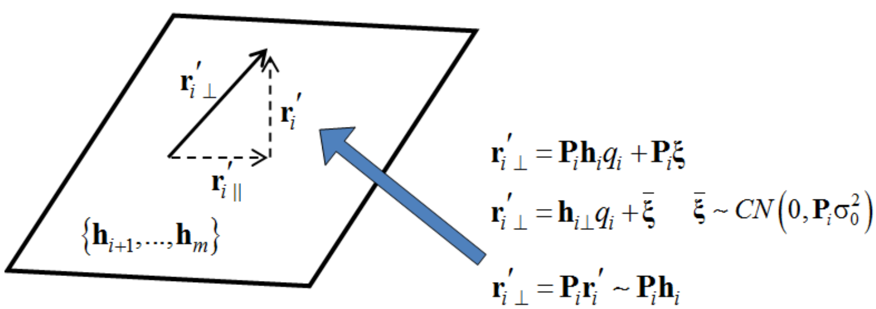

Figure 4: Geometric illustration of the interference nulling step.

After the interference nulling step, we are in presence of a vector channel under correlated noise:

$$\mathbf{r}_{i\perp}' = \mathbf{h}_{i\perp} q_i + \bar{\xi} \qquad (3.4)$$

where $\mathbf{h}_{i\perp} = \mathbf{P}_i \mathbf{h}_i$, $\bar{\xi} = \mathbf{P}_i \xi$ and $\bar{\xi}$ has the following distribution $\bar{\xi} \sim CN(0, \mathbf{P}_i \sigma_0^2)$. The correlation matrix of the after-projection noise $\bar{\xi}$ follows from the isotropic property of the

[5] $\mathbf{P}_i = \mathbf{I} - \mathbf{H}_i \left(\mathbf{H}_i^+ \mathbf{H}_i\right)^{-1} \mathbf{H}_i^+$ where $\mathbf{H}_i = [\mathbf{h}_{i+1}, ..., \mathbf{h}_m]$ [3]

unprojected noise ξ and the following property of the projection matrix: $\mathbf{P}_i = \mathbf{P}_i^+ = \mathbf{P}_i \mathbf{P}_i^+$. It was demonstrated in [16] that when applying MRC to the case of correlated noise, the noise correlation does not affect the output SNR and it is the same as in the case of i.i.d. noise after the projection. Moreover, it was shown that the optimum MRC weights already contain the projection matrix, and they can be called optimum ZF weights because they cancel interference and maximize the output SNR. The optimum ZF weight vector for detecting the *i*-th stream is given by

$$\mathbf{w}_i = \frac{\mathbf{h}_{i\perp}}{|\mathbf{h}_{i\perp}|} \tag{3.5}$$

and the output instantaneous SNR (conditional on no error at previous steps) is

$$\gamma_i = \frac{|\mathbf{h}_{i\perp}|^2}{\sigma_0^2} \tag{3.6}$$

Optimal ordering: The order in which transmitted symbols are detected is optimized to minimize the total error probability of the system. To change the symbol detection order is equivalent to re-ordering the columns of the channel matrix.

The detection ordering π_i is defined by: $\pi_i = \{i_1, i_2, ..., i_m\}$ where i_1 is the number of the stream which is detected first. For example, for a channel matrix given by $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2]$, ordering $\pi_i = \{2,1\}$ is equivalent to ordering the columns of the channel matrix as $\mathbf{H}' = [\mathbf{h}_2, \mathbf{h}_1]$. Assuming that the columns of the channel matrix $\mathbf{H}$ are ordered following the optimal ordering procedure i.e. $\mathbf{H}' = [\mathbf{h}'_1, \mathbf{h}'_2, ..., \mathbf{h}'_m]$, the system model can be expressed as

$$\mathbf{r} = \sum_i \mathbf{h}'_i q_i + \xi \tag{3.7}$$

After applying the optimum ZF weights at the *i*-th step the instantaneous after-processing SNR (conditional on no error at previous steps) is given by $\gamma_i' = |\mathbf{h}'_{i\perp}|^2 / \sigma_0^2$, where $\mathbf{h}'_{i\perp}$ is the projection of $\mathbf{h}'_i$ orthogonal to the subspace spanned by the $m-i$ remaining columns of the channel $\mathbf{H}'$. The after-processing instantaneous SNR at the *i*-th step is chi-squared distributed with $2(n-m+i)$ degrees of freedom i.e. $\gamma_i' = |\mathbf{h}'_{i\perp}|^2 / \sigma_0^2 \sim x^2_{2(n-m+i)}$, offering a diversity order of $(n-m+i)$ at each step. The diversity order increases from step to step with the first step having the lowest diversity order and the last step having the highest [19].

3.6. Optimum ordering in the uncoded V-BLAST

In the case of uncoded systems, the optimal order of stream processing is organized according to their after processing SNR in a decreasing order [3], that is, at each step the stream with the highest SNR at the output of the ZF detector will be detected first, and then its contribution is subtracted from the received vector. This process is repeated with the next strongest stream, among the remaining undetected ones. It was shown in [3] that this ordering strategy maximizes the minimum after-processing per-stream channel gain, i.e.

$$\pi^* = \arg\max_{\pi} \min_{i} |\mathbf{h}_{i\perp}(\pi)| \tag{3.8}$$

It can be shown that, when using two Tx antennas, this optimum ordering implies the detection of the stream with highest before detection channel gain first; i.e. the column with highest norm is placed first in the optimum ordering of the channel matrix columns [14].

Note: The above detection ordering was proposed by Foschini in [3]. Due to this fact, it will be referred to as Foschini ordering in this thesis.

Uncoded communication systems are not widely used; most modern communication systems use coding [33]. Therefore, it is important to discuss the optimal ordering for the coded V-BLAST. This issue is covered in the next chapters.

3.7. Summary

The detection algorithm used for the V-BLAST architecture is a combination of MMSE or ZF receiver with SIC. The ZF-VBLAST is a very attractive detection algorithm due to its potential and simplicity. The algorithm has three main procedures: interference cancellation, interference nulling and optimal ordering. The optimum ZF weights null out interference and maximize the output SNR.

4. Instantaneous Rate Allocation (IRA)

The optimum detection ordering for the coded V-BLAST under instantaneous rate allocation (IRA) is studied in this chapter. Under the IRA, the per-stream rates can be adjusted to match the per-stream channel capacity due to the use of capacity achieving codes. Furthermore, the power allocation is uniform across the streams, i.e. the normalized power assigned to each stream is equal to one.

After the interference cancellation and nulling steps, the equivalent scalar channel of the *i*-th stream is:

$$r_{out_i} = |\mathbf{h}_{i\perp}| q_i + \xi_i', \quad \xi_i' \sim CN(0, \sigma_0^2) \tag{4.1}$$

Equation (4.1) follows from applying the optimum ZF weight vector (3.5) to the vector channel in (3.4). The per-stream rate equals the per-stream capacity:

$$C_i = \ln\left(1 + |\mathbf{h}_{i\perp}|^2 \gamma_0\right) \qquad \text{[nat/s/Hz]} \tag{4.2}$$

where $\gamma_0 = 1/\sigma_0^2$ is the average SNR at each Rx antenna. The total capacity of the system is equal to the sum of the capacities of all the streams,

$$C = \sum_{i=1}^{m} C_i \tag{4.3}$$

The system is in outage when the total capacity is less than the system target rate mR and the outage probability is:

$$P_{out} = P\left[C < mR\right] = P\left[\sum_i C_i < mR\right] \tag{4.4}$$

From (4.4) it can be seen that an optimization strategy that improves the performance of the coded V-BLAST must target the outage probability or the total system capacity.

4.1. Optimum ordering under the IRA

4.1.1. General case

The next proposition states the optimum detection ordering strategy for the coded V-BLAST under the IRA.

Proposition 1: In the coded V-BLAST, under the IRA with capacity achieving temporal codes at each stream, the optimum detection ordering maximizes the instantaneous sum capacity of the system,

$$\pi^* = \arg\min_{\pi} P_{out}(\pi) = \arg\max_{\pi} C_{IRA}(\pi) = \arg\max_{\pi} \sum_{i=1}^{m} C_i(\pi) \tag{4.5}$$

where $P_{out}(\pi)$, $C(\pi)$ and $C_i(\pi) = \ln\left(1 + \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right)$ are the outage probability, the total system capacity and the per-stream capacity expressed as a function of the detection ordering π, and $\mathbf{h}_{i\perp}(\pi)$ is the projection of $\mathbf{h}_i$ orthogonal to the subspace spanned by the $m-i$ remaining columns of the channel matrix under ordering π.

Proof: The first equality in (4.5) follows because the optimal detection must minimize the outage probability, the second equality follows from the fact that the instantaneous optimizations of the capacity and outage probability achieve the same lowest value of the outage probability (the proof of this statement can be found in [32]). The third equality follows after noting that under the IRA the V-BLAST total capacity is equal to the sum capacity. ■

4.1.2. Two Tx antennas

When using two Tx antennas, an optimum detection order can be established based on the per-stream before-processing channel gains as stated in the following theorem.

Theorem 1: The optimum detection ordering for the coded V-BLAST with two Tx and n Rx antennas under the IRA is to detect the stream with highest before-detection channel gain at the last step[6]:

$$\pi^* = \arg\max_{\pi} \sum_{i=1}^{2} C_i(\pi) = \{1,2\} \quad \text{iff} \quad |\mathbf{h}_1| < |\mathbf{h}_2| \tag{4.6}$$

The "only if" part in (4.6) is true when $\varphi \neq \pi/2$, where φ is the angle between the two vector columns of the channel matrix. When $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ any ordering is optimum.

Proof: For the case of two Tx antennas, the channel matrix is given by $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2]$, there are two possible orderings: $\pi_1 = \{1,2\}$ and $\pi_2 = \{2,1\}$. The total capacities of these orderings are:

$$C_{IRA}(\pi_1) = \ln\left(1 + |\mathbf{h}_{1\perp 2}|^2 \gamma_0\right) + \ln\left(1 + |\mathbf{h}_2|^2 \gamma_0\right) \tag{4.7}$$

$$C_{IRA}(\pi_2) = \ln\left(1 + |\mathbf{h}_{2\perp 1}|^2 \gamma_0\right) + \ln\left(1 + |\mathbf{h}_1|^2 \gamma_0\right) \tag{4.8}$$

where $\mathbf{h}_{i\perp j}$ refers to the projection of $\mathbf{h}_i$ orthogonal to $\mathbf{h}_j$ and γ_0 is the same as in (4.2). Note that at the second step unprojected channel norms have been used since at this step the stream associated with the first column of the channel has already been detected and its interference was subtracted at the interference cancellation step. Therefore, the second stream is received without interference. To prove the "if" part, assume that ordering π_1 is optimum, such that

$$C_{IRA}(\pi_1) > C_{IRA}(\pi_2) \tag{4.9}$$

and the following chain of inequalities holds:

---

[6] This ordering is opposite of that for $(2\text{x}n)$ uncoded V-BLAST.

$$\left(1+\gamma_0\left|\mathbf{h}_1\right|^2\sin^2\varphi\right)\left(1+\left|\mathbf{h}_2\right|^2\gamma_0\right)>\left(1+\gamma_0\left|\mathbf{h}_2\right|^2\sin^2\varphi\right)\left(1+\left|\mathbf{h}_1\right|^2\gamma_0\right) \tag{4.10}$$

$$\Rightarrow\left|\mathbf{h}_2\right|^2+\left|\mathbf{h}_1\right|^2\sin^2\varphi>\left|\mathbf{h}_1\right|^2+\left|\mathbf{h}_2\right|^2\sin^2\varphi \tag{4.11}$$

$$\Rightarrow\left|\mathbf{h}_2\right|^2\left(1-\sin^2\varphi\right)>\left|\mathbf{h}_1\right|^2\left(1-\sin^2\varphi\right) \tag{4.12}$$

$$\Rightarrow\left|\mathbf{h}_2\right|^2>\left|\mathbf{h}_1\right|^2 \tag{4.13}$$

Inequality (4.10) follows from substituting (4.7) and (4.8) into (4.9), the geometric representation of the interference nulling step presented in [14], i.e.

$$\left|\mathbf{h}_{i\perp j}\right|=\left|\mathbf{h}_i\sin\varphi\right| \tag{4.14}$$

and the use of the following logarithmic identity:

$$\sum_{i=1}^{m}\ln\left(x_i\right)=\ln\left(\prod_{i=1}^{m}x_i\right) \tag{4.15}$$

Inequalities (4.11), (4.12) and (4.13) result from some straightforward mathematical manipulations. The “only if” part is proved by noting that the same chain of inequalities holds in the reverse direction i.e. starting in (4.13) and ending in (4.9). Equation (4.6) follows. Note that if $\varphi=\pi/2$ and/or $\left|\mathbf{h}_1\right|=\left|\mathbf{h}_2\right|$, the inequality (4.10) becomes an equality showing that any ordering is optimum under these circumstances. ■

4.1.3. General necessary optimality conditions

The aforementioned strategy is optimal for two Tx antennas. Also note that this strategy is SNR independent, i.e. as long as the stream with maximum before-detection channel gain is detected at the last step, the capacity of the system is maximized, independently of the SNR value. However, Monte-Carlo simulations show that for the case of $m>2$ the optimal detection ordering becomes SNR dependent.

Observation 1: The optimum detection ordering in the $(m\mathrm{x}n)$ coded V-BLAST $(m > 2)$ under the IRA is SNR dependent. In general, there is not a single detection ordering that is optimum at all SNR values.

A numerical example supporting Observation 1

The optimum detection orderings for the $(3\mathrm{x}3)$ system under the IRA were obtained for two different SNR values (-10dB and 0dB) given a Rayleigh fading channel realization. The capacities of these orderings at each SNR value are shown below.

The channel realization analysed is

$$\mathbf{H} = \begin{bmatrix} -0.5946 - 1.2503i & -0.8376 + 0.0962i & -1.2166 + 0.7775i \\ -0.0188 + 0.3720i & 0.7147 - 0.3839i & 0.1405 + 0.7745i \\ 0.1854 + 1.6291i & 0.9740 + 0.1342i & 1.1344 + 0.0909i \end{bmatrix}$$

Table 1: Performance of the optimal detection orderings under the IRA at SNR = -10dB and SNR = 0dB for the given channel realization.

	$C_{IRA \text{ at SNR}=-10\text{dB}}\left(\text{nats/sec/Hz}\right)$	$C_{IRA \text{ at SNR}=0\text{dB}}\left(\text{nats/sec/Hz}\right)$
$\pi^*_{IRA \text{ at SNR}=-10\text{dB}} = \{2,3,1\}$	0.48	2.44
$\pi^*_{IRA \text{ at SNR}=0\text{dB}} = \{1,2,3\}$	0.47	2.53

The example clearly shows the SNR dependency of the optimum ordering under the IRA when $m > 2$.

From Observation 1 it can be argued that, in general, a SNR independent detection ordering strategy (based only on **H**) cannot be optimum for all SNR values.

The strategy for $m > 2$ is to calculate the capacity of all possible orderings and select the one that with the maximum capacity. This increases the complexity and time consumption of the system due to many projections. Note that for m streams there are $m!$ possible detection orders – a quantity that increases very fast with m.

Based on the $(2\text{x}n)$ result, there are some necessary conditions (SNR independent) that must be satisfied by an optimum ordering. By exploiting these optimality conditions the number of possible orderings to evaluate capacity, and hence the complexity of the system, can be simplified.

Proposition 2: An optimum channel ordering must satisfy the following necessary conditions[7]:

$$\left|\mathbf{h}_{i-1\perp}\right| < \left|\mathbf{h}_{i\perp}\right| \quad \forall \;\; 2 \le i \le m \tag{4.16}$$

where $\mathbf{h}_{i-1\perp}$ and $\mathbf{h}_{i\perp}$ are the projections of vectors $\mathbf{h}_{i-1}$ and $\mathbf{h}_i$ orthogonal to the sub-space spanned by $\mathbf{h}_{i+1},...\mathbf{h}_m$.

Proof: Consider the following two orderings:

$$\pi_1 = \{1,...,i-1,i,...,m\} \tag{4.17}$$

$$\pi_2 = \{1,...,i,i-1,...,m\} \tag{4.18}$$

Note that the difference between these orderings is that the order at which two adjacent streams are detected has been swapped. Ordering π_1 is optimum provided that

$$C_{IRA}(\pi_1) > C_{IRA}(\pi_2) \tag{4.19}$$

If $\left|\mathbf{h}_{i-1\perp}\right| > \left|\mathbf{h}_{i\perp}\right| \Rightarrow C_{IRA}(\pi_1) < C_{IRA}(\pi_2)$, and π_1 cannot be the optimum ordering because there is another ordering that offers a higher capacity. This can be shown based on the following reasoning: the first $i-2$ and last $m-i$ streams of each ordering are the same, hence the streams associated with those columns have the same contribution in the total capacity of

[7] Proposition 2 holds given that all $\mathbf{h}_{i\perp}$ are of different length and non-orthogonal to each other. In the case where some $\mathbf{h}_{i\perp}$ are of equal length and/or orthogonal to each other, any ordering among them is optimum.

both orderings. The difference in capacity is determined by the detection ordering of the two swapped streams. The problem is reduced to the $(2\text{x}n)$ case where if $|\mathbf{h}_{i-1\perp}| > |\mathbf{h}_{i\perp}|$, from Theorem 1 the total capacities are compared as: $C_{IRA}(\pi_1) < C_{IRA}(\pi_2)$. ■

Example for m=3

In order to gain a better understanding of the necessary optimality conditions the case of three Tx antennas is discussed below. Assume that the detection ordering $\pi^* = \{1,2,3\}$ is optimum, i.e. $\pi^* = \arg\max_{\pi} \sum_{i=1}^{m} C_i(\pi) = \{1,2,3\}$.

Let us consider another ordering: $\pi_1 = \{1,3,2\}$.The difference in capacity between these orderings is given by the two last steps because the first stream is the same for both orderings and $|\mathbf{h}_{1\perp 2,3}| = |\mathbf{h}_{1\perp 3,2}|$. This reduces the problem to the $(2\text{x}n)$ case where the orderings are $\overline{\pi^*} = \{2,3\}$ and $\overline{\pi_1} = \{3,2\}$. From Theorem 1 it is known that the optimum strategy is to detect the stream associated to the highest unprojected channel gain at the last step, hence if $\pi^* = \{1,2,3\}$ provides the optimum detection ordering then $|\mathbf{h}_2| < |\mathbf{h}_3|$.

Now let us consider the detection ordering offered by $\pi_2 = \{2,1,3\}$ and compare its performance with that of the detection ordering given by $\pi^* = \{1,2,3\}$. It can be seen that the last stream is the same for both orderings. Given that at first and second steps the interference produced by the third stream must be projected out, the problem is reduced to the $(2\text{x}n)$ where the channel matrices under each ordering are given by $\mathbf{H}_{\pi^*} = [\mathbf{h}_{1\perp 3}\ \mathbf{h}_{2\perp 3}]$ and $\mathbf{H}_{\pi_2} = [\mathbf{h}_{2\perp 3}\ \mathbf{h}_{1\perp 3}]$. Again Theorem 1 states that the channel containing the column with highest unprojected norm at last position offers the highest capacity, hence if $\pi^* = \{1,2,3\}$ offers the optimum detection ordering then $|\mathbf{h}_{1\perp 3}| < |\mathbf{h}_{2\perp 3}|$. It can be concluded that: $\pi^* = \{1,2,3\}$ if $|\mathbf{h}_2| < |\mathbf{h}_3|$ and $|\mathbf{h}_{1\perp 3}| < |\mathbf{h}_{2\perp 3}|$. The same analysis can be extended to the case of arbitrary m.

Proposition 3: Three important conclusions follow from the necessary optimality conditions. Given that all $\mathbf{h}_{i\perp}$ are of different length and non-orthogonal to each other, the following holds.

- For a given order that meets the necessary optimality conditions, swapping two consecutive columns results in a new order that offers a lower capacity.
- A channel matrix under the optimum detection ordering will never contain the column with minimum norm at the last position.
- A channel matrix under the optimum detection ordering will never contain the column with maximum norm at the second last position.

There is, in general, more than one ordering that meet the necessary conditions for optimality. The best and worst case scenarios are considered below i.e. the minimum and maximum number of orderings that can meet the conditions.

Best case scenario

There are cases when only one ordering meets the necessary optimality conditions, the following proposition discusses such scenario.

Proposition 4: If the following properties of the channel matrix apply,

$$|\mathbf{h}_1| < |\mathbf{h}_2| < \ldots < |\mathbf{h}_m|, \qquad |\mathbf{h}_{i\perp}| < |\mathbf{h}_{j\perp}| \quad \forall \; i < j \tag{4.20}$$

where $\mathbf{h}_{i\perp}$ and $\mathbf{h}_{j\perp}$ are the projections of vectors $\mathbf{h}_i$ and $\mathbf{h}_j$ orthogonal to the sub-space spanned by the set of all remaining columns (not containing $\mathbf{h}_i$ and $\mathbf{h}_j$), then only one order can be optimum and hence it is SNR independent i.e. it is optimum for all SNR values.

Second part of (4.20) means that for any two columns, the column with higher index has a higher projected norm (projection orthogonal to the sub-space spanned by the set of columns not containing the selected ones).

As an example let us consider again the case of three Tx antennas. The number of possible detection orderings is $3! = 6$. Each possible detection ordering is represented by a different ordering of the columns of the channel matrix. The channel matrix under the ordering $\pi_1 = [1,2,3]$ is given by $\mathbf{H}_1 = [\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3]$. Let us assume that the following properties hold:

$$|\mathbf{h}_1| < |\mathbf{h}_2| < |\mathbf{h}_3| \tag{4.21}$$

$$|\mathbf{h}_{1\perp 3}| < |\mathbf{h}_{2\perp 3}| \tag{4.22}$$

$$|\mathbf{h}_{1\perp 2}| < |\mathbf{h}_{3\perp 2}| \tag{4.23}$$

$$|\mathbf{h}_{2\perp 1}| < |\mathbf{h}_{3\perp 1}| \tag{4.24}$$

where $\mathbf{h}_{i\perp j}$ is the projection of the *i*-th column orthogonal to the *j*-th column of the channel matrix. Note that (4.21) - (4.24) are explicitly the properties established in (4.20) for the case of $m = 3$, i.e. three Tx antennas. Based on the necessary optimality conditions, it can be seen that from property (4.21) the orderings $\pi_2 = [1,3,2]$, $\pi_3 = [2,3,1]$ and $\pi_4 = [3,2,1]$ can be eliminated; and from properties (4.22) and (4.23) the orderings $\pi_5 = [2,1,3]$ and $\pi_6 = [3,1,2]$ can be eliminated respectively. Therefore, only the ordering π_1 can be optimum.

Worst case scenario

It was already shown that in the best case scenario only one ordering meets the necessary optimality conditions. However, to gain more insights into these necessary optimality conditions it is important to consider also the worst case scenario. The worst case scenario can be defined as the maximum number of orderings that can meet the necessary optimality conditions for a given number of Tx antennas.

For a given number of Tx antennas (m), the number of necessary optimality conditions is equal to $(m\text{-}1)$. Each of the necessary optimality conditions can eliminate (on its own) half of the total number of combinations, but as they are not independent of each other, when applied all at the same time, different conditions may eliminate the same combinations and hence it is difficult to obtain an analytical result for the maximum number of combinations that are not eliminated after applying all the conditions. In light of this, the approach used here consisted of investigating the upper and lower bounds of this value. In this endeavour we rely on the following definition:

Definition 1: Independent necessary optimality conditions are defined as to those that are able to eliminate half of the combinations that were not eliminated after applying the previous conditions.

Now, based on the number of independent necessary optimality conditions the maximum number of orderings that can satisfy the necessary conditions can be bounded as stated in the following proposition:

Proposition 5: The maximum number of possible combinations (N) that can meet the necessary optimality conditions is bounded as follows

$$\frac{m!}{2^{m-1}} \le N \le \frac{m!}{2^{\lfloor m/2 \rfloor}} \tag{4.25}$$

where m is the number of Tx antennas, $(m-1)$ is the number of necessary optimality conditions and $\lfloor m/2 \rfloor$ is the number of independent necessary optimality conditions.

Proof: Each of the necessary optimality conditions can eliminate (on its own) half of the total $(m!)$ number of combinations. By applying only one condition, the number of orderings that are not eliminated is $m!/2$. If now a condition which is independent of the previous one is applied, the number of remaining orderings is $m!/2^2$. The first inequality in (4.25) follows from assuming that all the $(m-1)$ necessary optimality conditions are independent of each other. This a lower bound to N because all the necessary conditions are not independent. Based on many special cases, the number of independent necessary conditions is $\lfloor m/2 \rfloor$. The second inequality in (4.25) is the total number of orderings that are not eliminated taking into account only the independent conditions. This is clearly an upper bound because the number of orderings that are not eliminated by considering only the independent necessary conditions cannot be greater than the number of orderings that are not removed if all conditions are considered. ■

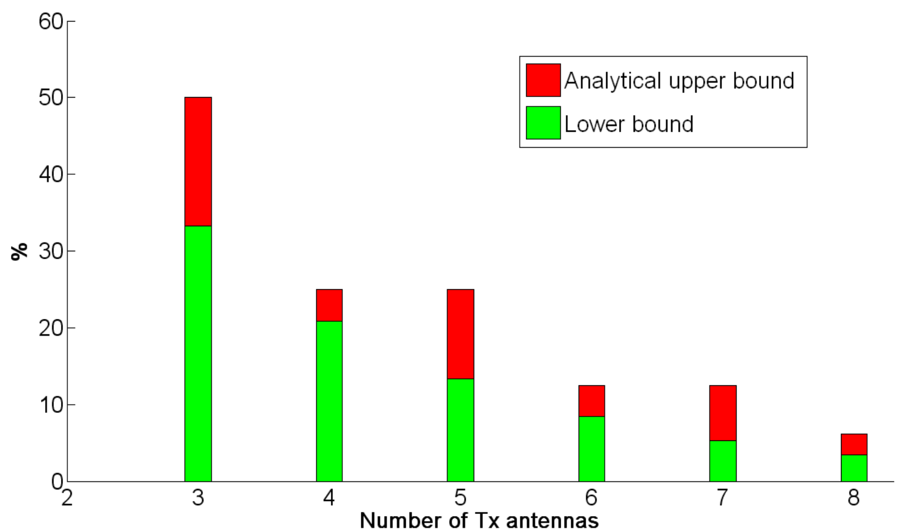

Figure 5: Upper bound given by (4.27) (dark grey) and numerical lower bound (light gray) of N (in percent) VS. # of Tx antennas.

Unfortunately, numerical simulations show that these analytical bounds for N are not very tight for large values of m. Based on a combinatorial Matlab code, it was found what we believe is the exact value of N for the $m = 3, 4, \ldots, 8$ cases. However, as the mathematical proof does not exist we can think of it as a numerical lower bound (tighter than the analytical lower bound in (4.25)). The graph in Figure 5 shows both the numerical lower bound and the analytical upper bound for N (expressed in percent) as a function of the number of Tx antennas. From both the graph presented in Figure 5 and Table 2 the following observation can be made:

Observation 2: The maximum number of detection orderings that satisfy the necessary optimality conditions represents only a small percentage of all the possible detection orderings, and this percentage decreases very fast as the number of Tx antennas is increased.

Table 2 illustrates the total number of possible detection orderings for a given number of Tx antennas and the number of detection orderings that satisfy the optimality conditions based on the proposed bounds. It can be seen how the detection orderings satisfying the conditions drop from a 50% for $m = 3$ to just 6.25% for $m = 8$ (based on the upper bound). Note that based on the numerical lower bound, the detection orderings that satisfy the necessary optimality conditions represent just the 3.4% of the total for the $m = 8$ case.

Table 2: Number of orderings remaining.

m	**Total comb.**	**Numerical lower bound**		**Analytical upper bound**	
		Orders remaining	**%**	**Orders remaining**	**%**
3	6	2	33.3	3	50
4	24	5	20.8	6	25
5	120	16	13.3	30	25
6	720	61	8.5	90	12.5
7	7040	272	5.4	880	12.5
8	40320	1385	3.4	2520	6.25

Figure 6 illustrates the reduction in complexity[8] offered by the analysis via the necessary optimality conditions (in the worst case scenario) as compared to the exhaustive

[8] in terms of the number of orderings which capacities need to be evaluated

search of the optimum ordering. It can be seen that although the complexity increases with the number of Tx antennas in both cases, the reduction in complexity by using the necessary optimality conditions also increases with *m*.

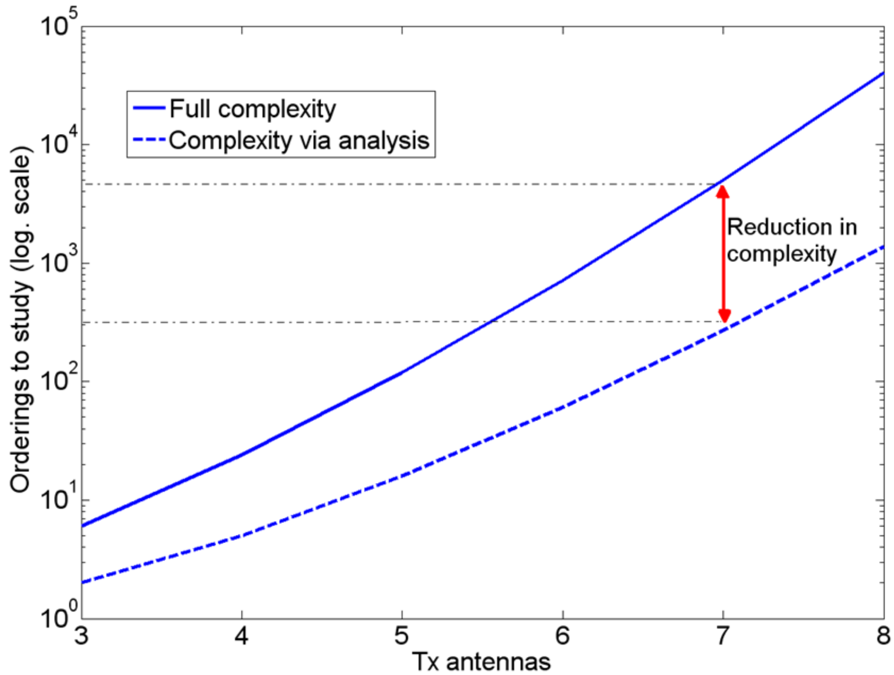

Figure 6: Complexity comparison between the exhaustive search of the optimum ordering and the analysis via the necessary optimality conditions.

Best and worst case scenarios (Numerical validations)

Monte-Carlo simulations were run in order to validate the previous results. Figure 7 shows the results of the numerical simulation for the $(4\text{x}4)$ coded V-BLAST under the IRA based on 10^6 i.i.d Rayleigh fading channel realizations. It can be seen from the graph that in a $(4\text{x}4)$ system, there is a relatively high probability (0.7) of having one detection ordering satisfying the necessary optimality conditions. The graph also shows that in the worst case only five detection orderings meet these conditions and this event takes place with very low probability $\left(3.2\text{x}10^{-5}\right)$. Note that the worst case shown in the simulation agrees exactly with the numerical lower bound of N for the case of four Tx antennas.

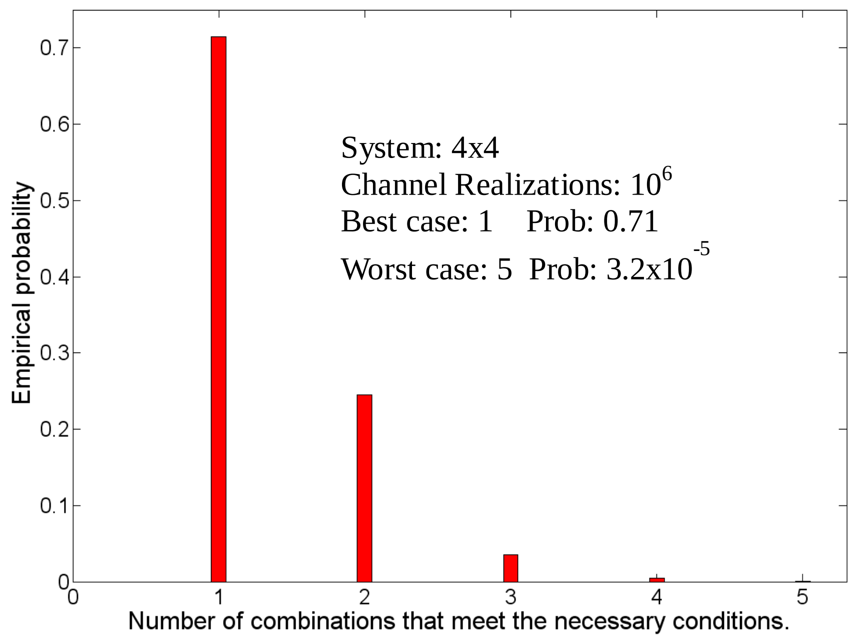

Figure 7: Normalized histogram of orderings that satisfy the necessary optimality conditions for the (4x4) system.

It is important to mention that for the case of $m \geq 5$ the numerical lower bound could not be achieved via Monte-Carlo simulations (see Figure 8), suggesting that as the number of Tx antennas is increased the worst case scenario is less probable. Therefore, the following observation can be made:

Observation 3: The worst case scenario takes place with very low probability and the probability of occurrence decreases as the number of Tx antennas is increased.

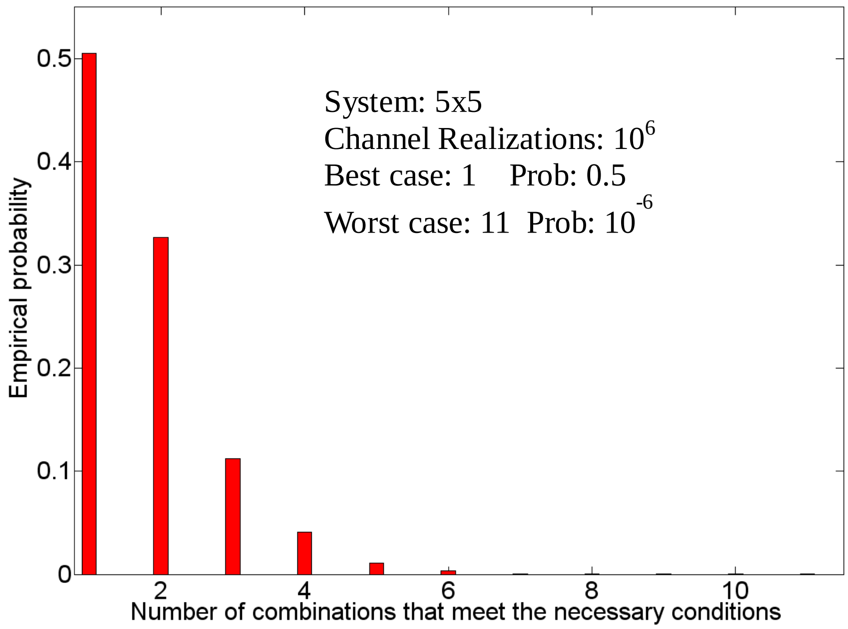

Figure 8: Normalized histogram of orderings that satisfy the necessary optimality conditions for the (5x5) system.

4.1.4. Optimal ordering strategies at low and high SNR

Approximated closed-form expressions for the optimal detection ordering strategies at low and high SNR are provided in this section. The expressions are obtained based on approximations of the per-stream capacity equation.

Low SNR approximation

The operating SNR of cellular systems with universal frequency reuse is typically very low, e.g. CDMA systems where the interference is considered to be part of noise. As V-BLAST can be also used in multiple access channels it is important to consider the optimum ordering strategy at low SNR.

Proposition 6: At low SNR the optimum detection ordering maximizes the sum of the after-processing channel gains

Proof: At low average SNR i.e. $\gamma_0 \ll 1$, for a given channel realization, the per-stream rate and the total capacity can be approximated respectively as:

$$C_i(\pi) = \ln\left(1 + \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right) \approx \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0 \quad \text{[nat/s/Hz]} \tag{4.26}$$

$$C_{IRA}(\pi) = \sum_{i=1}^{m} C_i(\pi) \approx \gamma_0 \sum_{i=1}^{m} \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \tag{4.27}$$

and the optimum detection ordering can be expressed as:

$$\max_{\{\pi\}}\left[\sum_{i=1}^{m} C_i(\pi)\right] = \gamma_0 \max_{\{\pi\}}\left[\sum_{i=1}^{m} \left|\mathbf{h}_{i\perp}(\pi)\right|^2\right] \tag{4.28}$$

Equation (4.26) and (4.27) result from the approximation $\ln(1+x) \approx x$ for small x. Equation (4.28) follows from the fact that the optimum detection ordering maximizes the system capacity. ■

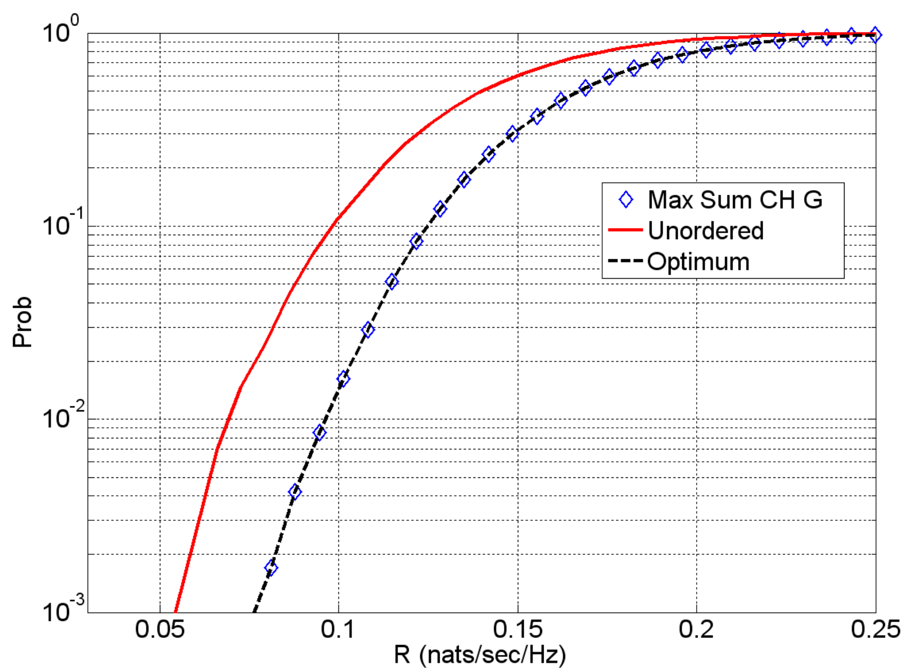

Figure 9: Empirical CDFs of the Max. Sum Ch. Gains detection ordering, the optimum detection ordering and the unordered detection for the (5x5) system; SNR=-20dB; 10^4 channel realizations.

In order to validate proposition 6, extensive Monte-Carlo simulations were undertaken. Figures 9-10 show the empirical cumulative distribution functions (CDFs) of the optimum detection ordering, the detection ordering that maximizes the after-processing channel gains and the unordered detection for $(5x5)$ and $(4x4)$ V-BLAST systems respectively. These CDFs are based on based on 10^4 i.i.d. Rayleigh fading channel realizations. It can be seen that the detection ordering that maximizes the after-processing channel gains is in fact optimum at low SNR.

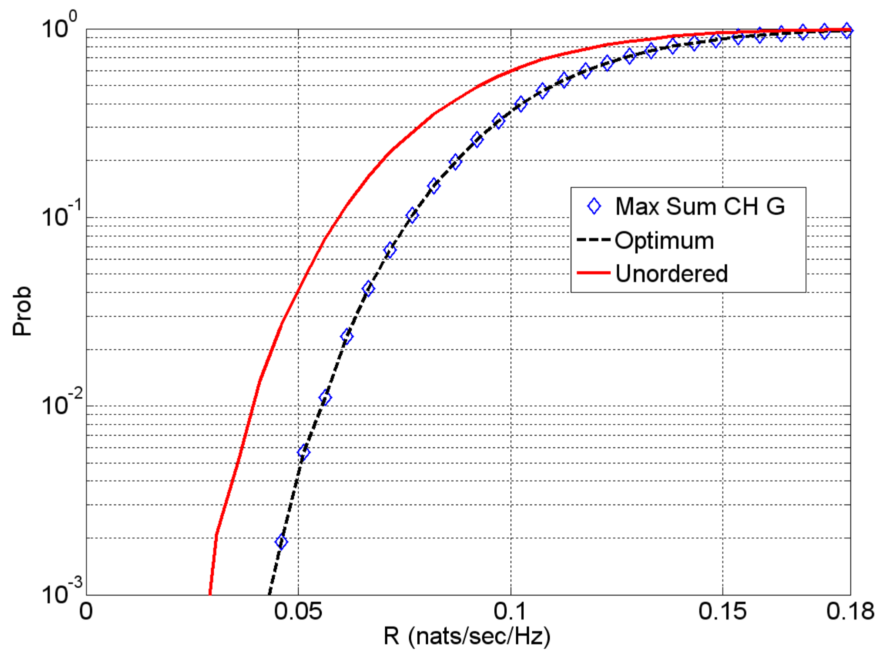

Figure 10: Empirical CDFs of the Max. Sum Ch. Gains detection ordering, the optimum detection ordering and the unordered detection for the (4x4) system; SNR=-20dB; 10^4 channel realizations.

High SNR approximation

The high SNR optimum detection ordering can be found in a similar way as it was found at low SNR. A key question in this context is whether the optimum detection ordering offers a significant advantage at high SNR? This issue is investigated below.

Proposition 7: At high SNR all channel orders provide approximately the same sum capacity, hence the optimal ordering does not offer a significant improvement to the system.

Proof: At high SNR i.e. $\gamma_0 \gg 1$, for a given channel realization, the per-stream rate and the total capacity can be approximated respectively as:

$$C_i(\pi) = \ln\left(1 + \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right) \approx \ln\left(\left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right) \quad (4.29)$$

$$C_{IRA}(\pi) = \sum_{i=1}^{m} C_i(\pi) \approx \mathrm{m}\ln(\gamma_0) + \ln\left(\prod_{i=1}^{m}\left|\mathbf{h}_{i\perp}(\pi)\right|^2\right) \quad (4.30)$$

and the optimum detection ordering can be expressed as:

$$\begin{aligned}\max_{\{\pi\}}\left[\sum_{i=1}^{m} C_i(\pi)\right] &= \mathrm{m}\ln(\gamma_0) + \max_{\{\pi\}}\left[\ln\left(\prod_{i=1}^{m}\left|\mathbf{h}_{i\perp}(\pi)\right|^2\right)\right] \\ &= \mathrm{m}\ln(\gamma_0) + \ln\left(\left|\mathbf{H}\mathbf{H}^{+}\right|\right)\end{aligned} \quad (4.31)$$

Equation (4.29) follows from the approximation $\ln(1+x) \approx \ln(x)$ for large x. Equation (4.30) is obtained from the logarithmic identities (4.15) and

$$\ln\left(x^m\right) = m\ln(x) \quad (4.32)$$

The first equality in (4.31) follows from the facts that the optimum detection ordering maximizes the system capacity while the second equality follows from

$$\prod_{i=1}^{m}\left|\mathbf{h}_{i\perp}(\pi)\right|^2 = \left|\mathbf{H}\mathbf{H}^{+}\right| \quad \forall\pi \quad (4.33)$$

Proof of equation (4.33) can be found in [34]. ■

Monte-Carlo simulations were carried out in order to validate the analytical results supporting Proposition 7. Figure 11 shows the empirical curve of outage probability as a function of SNR of the optimum detection ordering and the unordered detection for the $(5\text{x}5)$ V-BLAST system based on 10^3 i.i.d. Rayleigh fading channel realizations for a target rate of 40 nats/sec/Hz. It can be seen that both detection ordering strategies offer the same performance.

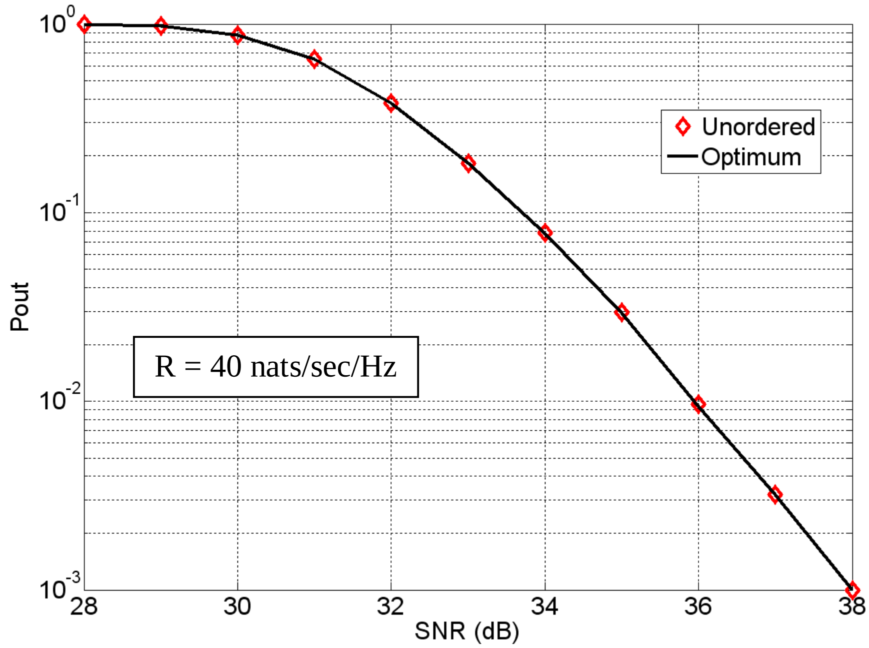

Figure 11: Empirical Pout vs. SNR of the optimum detection ordering and the unordered detection for the (5x5) system at a high target rate based on 10^4 channel realizations.

4.2. Suboptimal orderings

In this section suboptimal orderings that perform very close to the optimum one are proposed.

Inverse ordering

This ordering strategy follows the same principle as the optimum ordering strategy for uncoded systems proposed by Foschini in [3], but the stream with highest after-processing SNR is detected last (as a guiding principle). The algorithm is as follows,

1- Select the stream with largest $\left|\mathbf{h}_{i_1}\right|$ and detect.

2- Remove the detected stream, select the one with the largest $\left|\mathbf{h}_{i_2 \perp i_1}\right|$ and detect.

3- Repeat step 2 until finish.

In this way, the last column of the channel matrix is the one with maximum norm. Note that at the last step of the V-BLAST algorithm it is not necessary to project out any interference.

The second last column will be the one with highest projected norm (projection orthogonal to the column with highest norm), and so on up to the first column. It is evident that, under this ordering, the necessary optimality conditions are always satisfied.

This is a SNR independent ordering strategy (just based on $\mathbf{H}$). It is known from Observation 1 that for $m \geq 3$ the optimum detection ordering is SNR dependent. Hence, in general, the inverse ordering cannot be optimum. However, as illustrated in Figures 12-13, this ordering performs very close to the optimum one.

Max Sum Capacity of the Last Two Steps (MSCL2S) ordering

Based on the fact that the total capacity of the system is dominated by the last two steps of the V-BLAST algorithm, the MSCL2S ordering is also proposed. Under this ordering strategy, the last column of the channel matrix is selected such that the capacity of the last two steps of the V-BLAST is maximized. The remaining columns are ordered following the inverse ordering criteria. In this case the necessary optimality conditions are also satisfied. First notice that the last column is selected based on the capacity offered by the last two steps, and from Theorem 1 the last two columns will satisfy $|\mathbf{h}_{m-1}| \leq |\mathbf{h}_m|$. Second, as the rest of the channel matrix are ordered following the inverse ordering criteria, the remaining $m-2$ conditions are also satisfied.

Note that this ordering strategy is SNR dependent. The last column of the channel matrix is chosen based on the capacity equation of the two last steps (the capacity equation is SNR dependent) and the remaining columns are ordered depending on the last one. Despite this, in general, it cannot be the optimum. This is because the fact that selecting the last column such that the capacity of the last two steps is maximized does not guarantee that the total system capacity is maximized as well. However, the capacity offered by this ordering is also very close to the optimum one at all SNR values. The performance of this detection ordering strategy is illustrated in Figures 12-13.

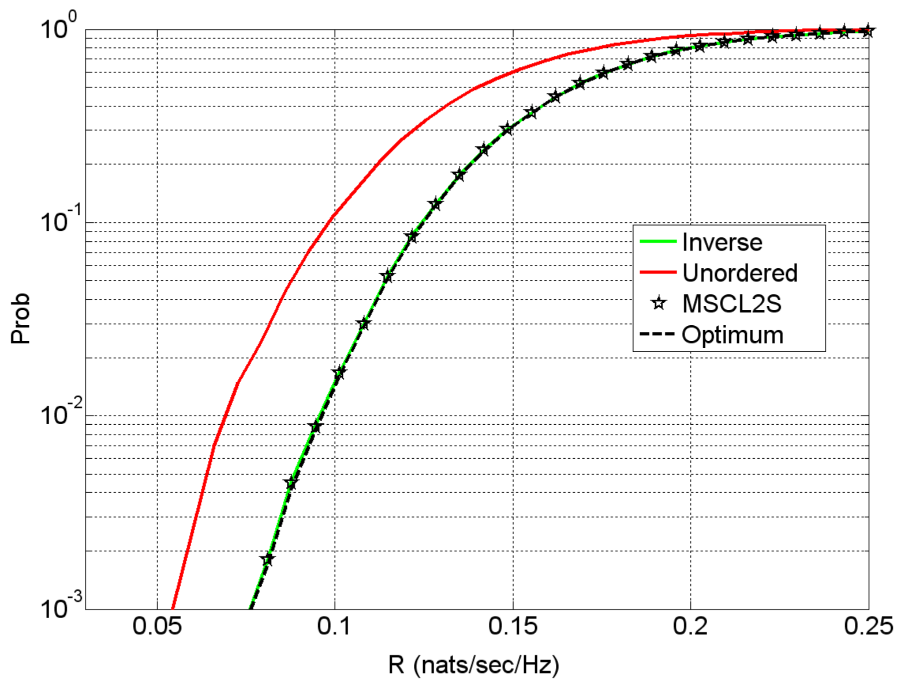

Figure 12: Empirical CDFs of the inverse ordering, the MSCL2S ordering, the optimum ordering and the unordered detection for the (5x5) system; SNR=-20dB; 10^4 channel realizations.

Figures 12-13 show the empirical CDF curves, based on 10^4 i.i.d. Rayleigh fading channel realizations, of the proposed suboptimal detection orderings, the optimum detection ordering, and the unordered detection for the $(5x5)$ and the $(4x4)$ coded V-BLAST systems respectively at different SNR values. It is seen that both suboptimal detection orderings offer an almost optimum performance.

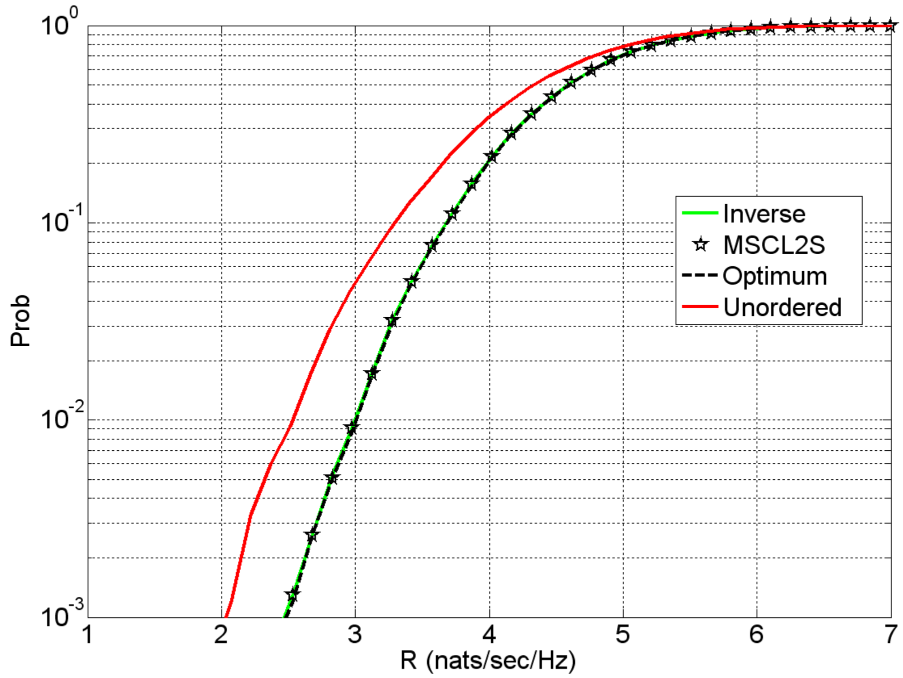

Figure 13: Empirical CDFs of the inverse ordering, the MSCL2S ordering, the optimum ordering and the unordered detection for the (4x4) system; SNR=0dB; 10^4 channel realizations.

Unprojected ordering

In this detection ordering strategy the streams are detected based on their respective unprojected channel gains in an increasing order. The stream associated to the channel column with minimum norm is detected first and the one associated to the channel column with maximum norm is detected last. This is the simplest strategy since it is not necessary to make any projection in order to set up the stream detection order. From Theorem 1 the unprojected detection ordering strategy is optimal for the (2 x n) coded V-BLAST under the IRA and Monte-Carlo simulations show that it offers a good performance at all SNR values for small number of Tx antennas i.e. $m = 3,4,5$ (see Figures 14-15). However, its performance degrades when the number of Tx antennas is increased.

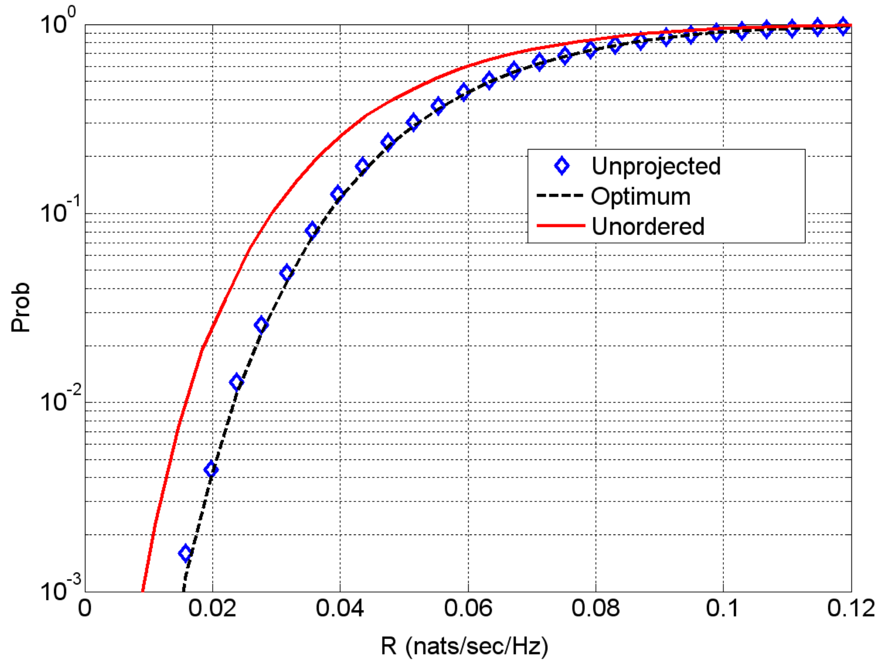

Figure 14: Empirical CDFs of the unprojected detection ordering, the optimum detection ordering and the unordered detection for the (3x3) system; SNR=-20dB; 10^4 channel realizations.

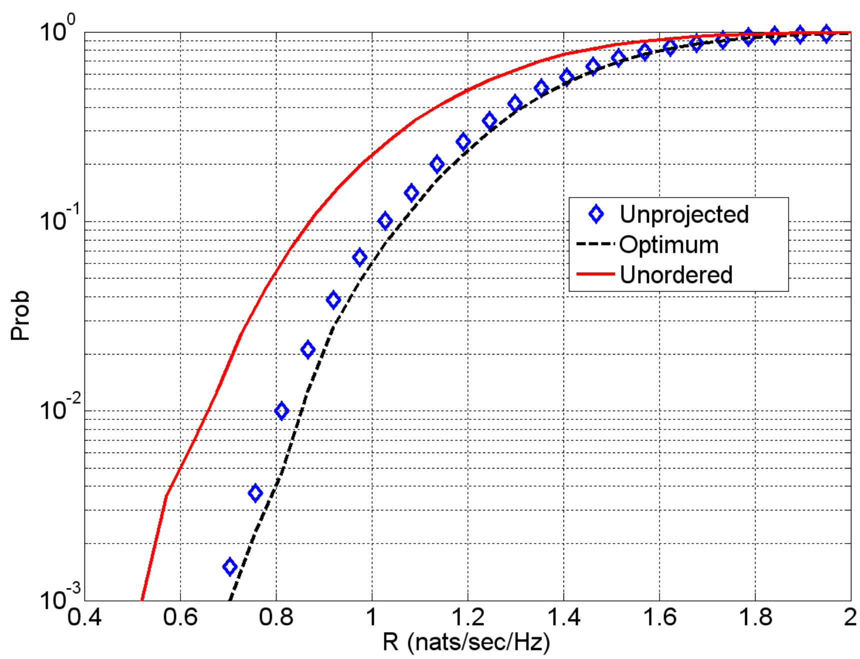

Figure 15: Empirical CDFs of the unprojected detection ordering, the optimum detection ordering and the unordered detection for the (5x5) system; SNR=-10dB; 10^4 channel realizations.

4.3. Summary

The optimum detection ordering for the coded V-BLAST under the IRA was studied. Closed-form approximations were derived for the optimum ordering at low and high SNR. The optimum ordering maximizes the sum of the after-processing channel gains at low SNR while all orderings offer approximately the same performance at high SNR.

For the case of two Tx antennas, the optimum ordering can be established based on the before-processing channel gains: the stream with highest before-processing channel gain must be detected last. For the case of $m > 2$, the optimum ordering becomes SNR dependent. Based on the $(2\text{x}n)$, SNR independent necessary optimality conditions were derived for arbitrary number of Tx antennas. A small percentage of orderings satisfy the necessary conditions for optimality and this percentage decreases with the number of Tx antennas. The minimum number of orderings that can satisfy the necessary optimality conditions is one and this scenario takes place with high probability. The maximum number of orderings that can satisfy these conditions is still an open problem. In this chapter, lower and upper bounds of this maximum were determined. Simulations demonstrate that the numerical lower bound is reached with very low probability.

Suboptimum detection orderings were proposed. Both the inverse and the MSCL2S orderings offer an almost optimum performance for arbitrary number of Tx antennas. The unprojected ordering is the simplest strategy and its performance is very good for a small number of Tx antennas. However, its performance degrades with the number of Tx antennas.

5. Uniform Power and Rate Allocation (URA)

In the previous chapter, an instantaneous rate allocation at each stream was assumed i.e. the use of a non-uniform transmission rate among the streams where the per-stream rate is adjusted to match the per-stream capacity for each channel realization. The system with the uniform power and rate allocation (URA) among the streams is considered in this chapter. This scenario takes place, for example, when the same modulation/coding is used at each stream to simplify the system design.

In the presence of the URA, the system is in outage when at least one stream cannot support the target rate *R*. Hence, the Tx rate at each stream is equal to the capacity of the weakest stream in order to attain reliable communication. The total capacity of the system is:

$$C_{URA} = mC_{\min} \tag{5.1}$$

where $C_{\min}$ is the capacity of the weakest stream and is expressed as

$$C_{\min} = \min_i C_i = \ln\left(1 + \min_i |\mathbf{h}_{i\perp}|^2 \gamma_0\right) \tag{5.2}$$

where C_i is as in (4.2). The second equality in (5.2) follows from the fact that $\ln(x)$ is an increasing function of x.

5.1. Optimum ordering under the URA

5.1.1. General case

We are now in a position to establish the optimum ordering strategy for the coded V-BLAST under the URA.

Proposition 8: The optimum ordering in the coded V-BLAST under the URA with capacity achieving temporal codes at each stream is given by:

$$\pi^* = \arg\max_{\pi} \min_{i} C_i(\pi) = \arg\max_{\pi} \min_{i} \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \tag{5.3}$$

where $C_i(\pi)$ and $\mathbf{h}_{i\perp}(\pi)$ are as in (4.5).

Proof: The system capacity under the URA can be expressed as a function of the detection ordering,

$$C_{URA}(\pi) = m \min_{i} C_i(\pi) = m \ln\left(1 + \min_{i} \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right) \tag{5.4}$$

and the following set of equations for the optimum detection ordering (π^*) holds

$$\pi^* = \arg\max_{\pi} \min_{i} C_i(\pi) \tag{5.5}$$

$$= \arg\max_{\pi} m \log\left(1 + \min_{i} \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right) \tag{5.6}$$

$$= \arg\max_{\pi} \min_{i} \left|\mathbf{h}_{i\perp}(\pi)\right|^2 \tag{5.7}$$

Equation (5.5) follows from the fact that the optimum detection ordering must maximize the instantaneous system capacity. Equation (5.6) results from substituting (5.4) into (5.5) and equation (5.7) follows from the fact that $\ln(x)$ is an increasing function of x. ■

The optimum ordering under the URA can be interpreted as the one that maximizes the after-processing channel gain (SNR) of the weakest stream.

From equation (5.3) three more propositions follow.

Proposition 9: In the coded V-BLAST under the URA and using capacity achieving temporal codes at each stream, the optimum detection ordering is SNR independent, i.e. it depends on $\mathbf{H}$ but not on γ_0. Hence, for a given channel realization, an optimum detection ordering remains optimum for all SNR values.

Proposition 10: In the coded V-BLAST under the URA and using capacity achieving temporal codes at each stream, there is, in general, more than one optimum order i.e. different orders may offer the same maximum capacity.

The main idea under Proposition 10 is that the optimum ordering strategy depends only on the after-processing channel gain of the weakest stream. The columns to the left or right of the column offering the minimum stream capacity (in the optimum ordering) can be reordered and, as long as the minimum stream remains the same, the orderings generated as a result of the columns permutation process achieve the same maximum capacity and hence they are optimal. One simple example is when $\mathbf{H}$ is diagonal, in this case any ordering is optimum. Proposition 10 is also true for non-diagonal $\mathbf{H}$ as proved below.

Proof: Proposition 10 is proved for the case of $m = 3$ and it can be easily extended to arbitrary m.

Let us consider the case of three Tx antennas $(m = 3)$. Without loss of generality it can be assumed that the ordering $\pi^* = \{1^*, 2, 3\}$ is optimum, i.e.

$$\pi^* = \arg\max_{\pi} \min_{i} C_i(\pi) = \{1^*, 2, 3\} \tag{5.8}$$

where 1^* means that the first step of this ordering has the minimum capacity i.e.

$$1^* \Rightarrow \min_{i}\left[C_1, C_2, C_3\right] = C_1 \Rightarrow \left|\mathbf{h}_{1\perp 2,3}\right|^2 \le \left|\mathbf{h}_{2\perp 3}\right|^2, \left|\mathbf{h}_3\right|^2 \tag{5.9}$$

where C_i is referring to the capacity of the *i*-th stream of the V-BLAST algorithm. Now assume ordering $\pi_1 = \{1^*, 3, 2\}$ where

$$1^* \Rightarrow \min_{i}\left[C_1, C_2, C_3\right] = C_1 \Rightarrow \left|\mathbf{h}_{1\perp 3,2}\right|^2 \le \left|\mathbf{h}_{3\perp 2}\right|^2, \left|\mathbf{h}_2\right|^2 \tag{5.10}$$

The capacities of both orderings are equal, i.e.

$$C_{URA}(\pi^*) = C_{URA}(\pi_1) = 3\log\left(1 + \left|\mathbf{h}_{1\perp 2,3}\right|^2 \gamma_0\right) \tag{5.11}$$

and hence both orderings are optimal. ■

Proposition 11: Since Foschini ordering maximizes the minimum stream channel gain, it is optimum for the coded V-BLAST under the URA. Two illustrative examples of this statement can be seen in Figures 16-17.

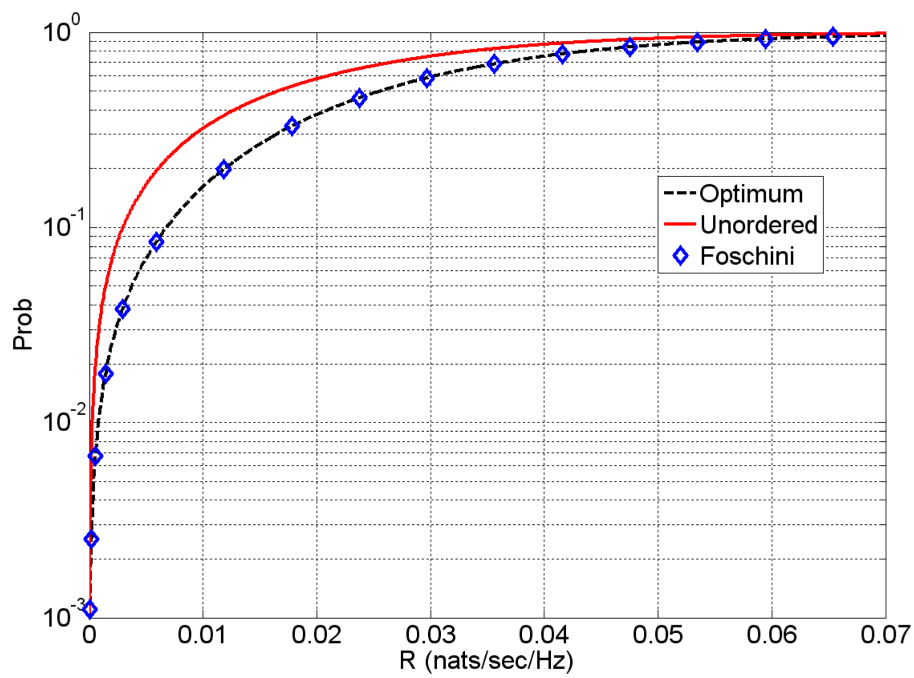

Figure 16: Empirical CDFs of the Foschini detection ordering, the optimum detection ordering and the unordered detection under the URA for the (3x3) system; SNR=-20dB; 10^5 channel realizations.

Numerical simulations were undertaken to validate proposition 11. Figures 16-17 show the empirical CDFs, based on 10^5 i.i.d. Rayleigh fading channel realizations, of the Foschini detection ordering, the optimum detection ordering, and the unordered detection for the $(3\text{x}3)$ V-BLAST system under the URA at the SNR values of -20dB and 20dB respectively. It is shown both the optimality of the Foschini detection ordering and the improvement offered by the optimum ordering process in the coded V-BLAST under the URA.

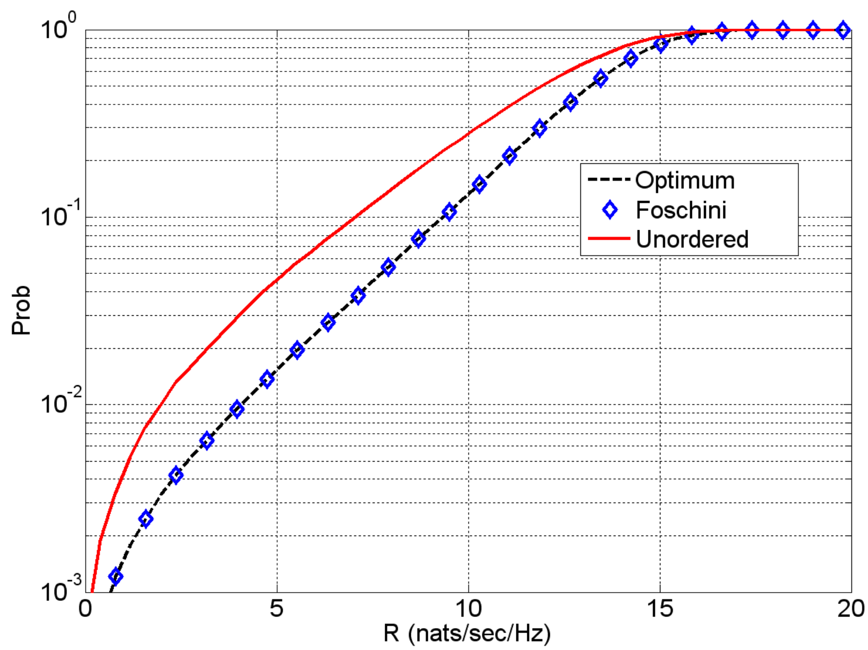

Figure 17: Empirical CDFs of the Foschini detection ordering, the optimum detection ordering and the unordered detection under the URA for the (3x3) system; SNR=20dB; 10^5 channel realizations.

5.1.2. Two Tx antennas

Similarly to the case of the IRA, the optimum detection ordering strategy under the URA for the case of two Tx antennas can be established based on the per-stream before-processing channel gains as stated in the following theorem.

Theorem 2: The optimum detection ordering for the coded V-BLAST with two Tx and n Rx antennas under the URA is to detect the stream with highest before-detection channel gain first:

$$\pi^* = \arg\max_{\pi} \min_{i} C_i(\pi) = \{1,2\} \quad \text{iff} \quad |\mathbf{h}_1| > |\mathbf{h}_2| \quad \text{for } \forall \varphi \neq 0 \qquad (5.12)$$

The "only if" part in (5.12) is true when $\varphi \neq \pi/2$. When $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ any ordering is optimum.

Proof: For the case of two Tx antennas, the channel matrix is given by $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2]$. To prove the "if" part, without loss of generality, it can be further assumed that $|\mathbf{h}_1| > |\mathbf{h}_2|$. There are two possible orderings: $\pi_1 = \{1,2\}$ and $\pi_2 = \{2,1\}$. The total capacities are:

$$C_{URA}(\pi_1) = 2\ln\left(1 + \min\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right]\gamma_0\right) \tag{5.13}$$

$$C_{URA}(\pi_2) = 2\ln\left(1 + \min\left[|\mathbf{h}_{2\perp 1}|^2, |\mathbf{h}_1|^2\right]\gamma_0\right) \tag{5.14}$$

Equations (5.13) and (5.14) follow from (5.4) and the appropriate substitution of the channel gains depending on the specific ordering. The following chain of equalities and inequalities holds:

$$|\mathbf{h}_1|^2 > |\mathbf{h}_{1\perp 2}|^2 = |\mathbf{h}_1|^2 \sin^2\varphi > |\mathbf{h}_2|^2 \sin^2\varphi$$
$$= |\mathbf{h}_{2\perp 1}|^2 \tag{5.15}$$
$$|\mathbf{h}_2|^2 > |\mathbf{h}_{2\perp 1}|^2$$

$$\Rightarrow \min\left[|\mathbf{h}_{2\perp 1}|^2, |\mathbf{h}_1|^2\right] = |\mathbf{h}_{2\perp 1}|^2 < \min\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right] \tag{5.16}$$

$$\Rightarrow C_{URA}(\pi_1) > C_{URA}(\pi_2) \tag{5.17}$$

Both equalities in (5.15) follow from the use of the geometrical representation (4.14). The first and third inequalities in (5.15) hold because $\sin^2\varphi < 1$[9] and the second inequality result from $|\mathbf{h}_1| > |\mathbf{h}_2|$. The equality and the inequality in (5.16) follow immediately from (5.15) and finally (5.17) follows from (5.13), (5.14) and (5.16).

To prove the "only if" part, assume that $C_{URA}(\pi_1) > C_{URA}(\pi_2)$ so that

$$\min\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right] > \min\left[|\mathbf{h}_{2\perp 1}|^2, |\mathbf{h}_1|^2\right] \tag{5.18}$$

Now there are two options. The first option is:

---

[9] Assuming that $\varphi \neq \pi/2$.

$$\min\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right] = |\mathbf{h}_2|^2 \tag{5.19}$$

so that

$$|\mathbf{h}_2|^2 \le |\mathbf{h}_{1\perp 2}|^2 < |\mathbf{h}_1|^2 \tag{5.20}$$

where the first inequality in (5.20) results from (5.19) and the second inequality follows because the projection of a vector cannot increase its norm. The second option is:

$$\min\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right] = |\mathbf{h}_{1\perp 2}|^2 \tag{5.21}$$

and thus the following inequalities hold:

$$|\mathbf{h}_{1\perp 2}|^2 = \min\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right] > \min\left[|\mathbf{h}_{2\perp 1}|^2, |\mathbf{h}_1|^2\right] \tag{5.22}$$

$$\Rightarrow |\mathbf{h}_{1\perp 2}|^2 > |\mathbf{h}_{2\perp 1}|^2 \tag{5.23}$$

$$\Rightarrow |\mathbf{h}_1|^2 \sin^2\varphi > |\mathbf{h}_2|^2 \sin^2\varphi \tag{5.24}$$

$$\Rightarrow |\mathbf{h}_1|^2 > |\mathbf{h}_2|^2 \tag{5.25}$$

The inequality in (5.22) results from (5.21) and (5.18). Inequality (5.23) follows from (5.22) and $|\mathbf{h}_1|^2 > |\mathbf{h}_{1\perp 2}|^2$. Inequality (5.24) follows from the geometrical representation (4.14). Equation (5.12) follows. Note that if $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ the inequality in (5.16) becomes an equality and thus any ordering is optimum. If $\varphi = 0$, the column vectors of the channel matrix are parallel. Therefore, the length of the projected vector is zero and thus the transmission strategy is to keep only one active stream (the stream with highest before-detection channel gain). ■

5.2. Summary

The optimum detection ordering for the coded V-BLAST under the URA was studied. The optimum detection ordering is SNR independent and it maximizes the minimum after-processing stream gain. It was shown that the Foschini ordering is optimum for arbitrary number of Tx antennas; however, several orderings may offer the same optimum performance when $m \geq 3$, i.e. the optimum ordering is not unique.

For the case of two Tx antennas, the optimum ordering can be established based on the before-processing channel gains: the stream with highest before-processing channel gain must be detected first.

6. Non-uniform power allocation

Uniform power allocation at each stream was assumed in the preceding analysis, i.e. the normalized power assigned to each stream is equal to one. It was demonstrated in [32] that by allocating different powers to each Tx antenna the coded V-BLAST performance is improved. This chapter studies the optimal ordering strategies for the coded V-BLAST under the instantaneous power allocation (uniform rate) and under the instantaneous power/rate allocation. The following figure provides the block diagram of the system.

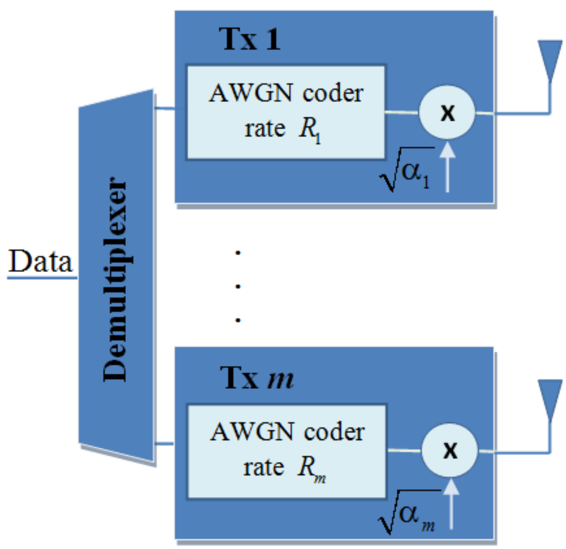

Figure 18: Tx side architecture of the coded V-BLAST with non-uniform power allocation. The total normalized power is $\sum_{i=1}^{m}\alpha_i = m$.

The system model is given by:

$$\mathbf{r} = \mathbf{H\Lambda q} + \xi = \sum_i \mathbf{h}_i \sqrt{\alpha_i} q_i + \xi \tag{6.1}$$

where **q**, **r**, **ξ**, and **H** are the same as in (3.1). $\mathbf{\Lambda}$ is a diagonal matrix whose entries represent the squared root of the power assigned to each stream. After the interference cancellation and nulling steps, the equivalent scalar channel of the *i*-th stream is:

$$r_{out_i} = |\mathbf{h}_{i\perp}| \sqrt{\alpha_i} q_i + \xi_i', \quad \xi_i' \sim CN(0, \sigma_0^2) \tag{6.2}$$

Given that capacity achieving temporal codes are used at each stream, each stream can support a rate equal to its instantaneous capacity given by

$$C_i = \ln\left(1 + \alpha_i |\mathbf{h}_{i\perp}|^2 \gamma_0\right) \qquad \text{[nat/s/Hz]} \tag{6.3}$$

where $\mathbf{h}_{i\perp}$ and γ_0 are the same as in (4.2) and α_i is the power allocated to stream $\boldsymbol{i}$.

Furthermore, the power allocation can be chosen in an optimum way. Any optimum instantaneous power allocation must target the outage probability or the total capacity of the system.

6.1. Instantaneous Power Allocation (IPA)

The optimum instantaneous power allocation strategy, subject to the total power constraint, when per-stream rates are equal to the minimum per-stream capacity is discussed in this section. Based on the previous specification, the per-stream rate under the IPA is given by:

$$R_i = C_{\min} = \min_i \ln\left(1 + \alpha_i |\mathbf{h}_{i\perp}|^2 \gamma_0\right) \tag{6.4}$$

Then for a given power allocation vector $\boldsymbol{\alpha}$ the system capacity under the IPA is:

$$C(\boldsymbol{\alpha}) = m \min_i \ln\left(1 + \alpha_i |\mathbf{h}_{i\perp}|^2 \gamma_0\right) \tag{6.5}$$

and the optimization problem for a given order π can be formulated as:

$$\begin{aligned} C_{IPA}(\pi) &= m \max_{\boldsymbol{\alpha}(\pi)} \min_i \ln\left(1 + \alpha_i(\pi) |\mathbf{h}_{i\perp}(\pi)|^2 \gamma_0\right) \\ \text{s.t.} \quad & \sum_i \alpha_i(\pi) = m, \quad \alpha_i(\pi) \geq 0 \end{aligned} \tag{6.6}$$

The solution to this optimization problem is the channel inversion [32]. Under the channel inversion strategy, more power is allocated to the stream with smaller instantaneous channel gain. This power allocation is given by

$$\alpha_i(\pi) = \overline{g}(\pi) / g_{i\perp}(\pi) \tag{6.7}$$

where $g_{i\perp}(\pi) = \left|\mathbf{h}_{i\perp}(\pi)\right|^2$ and $\overline{g}(\pi)$ is the harmonic mean per-stream power gain for a given ordering,

$$\overline{g}(\pi) = \left(\frac{1}{m}\sum_i \left(g_{i\perp}(\pi)\right)^{-1}\right)^{-1} \tag{6.8}$$

and the maximum capacity for a given ordering is:

$$C_{IPA}(\pi) = \begin{cases} m\ln\left(1+\overline{g}(\pi)\gamma_0\right), & g_{i\perp}(\pi) > 0 \ \ \forall i \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

The explicit derivations to obtain (6.7) and (6.9) can be found in [32].

6.1.1. Optimum ordering under the IPA

General case

We are now in a position to establish the optimum ordering strategy for the coded V-BLAST under the IPA.

Proposition 12: The optimum ordering in the coded V-BLAST under the IPA with capacity achieving temporal codes at each stream is given by:

$$\pi^* = \arg\max_{\pi} \overline{g}(\pi) \tag{6.10}$$

Proof: The following set of equalities proves (6.10):

$$\pi^* = \arg\max_{\pi} C_{IPA}(\pi) \tag{6.11}$$

$$= \arg\max\left[m\ln\left(1+\overline{g}(\pi)\gamma_0\right)\right] \tag{6.12}$$

$$= \arg\max_{\pi} \overline{g}(\pi) \tag{6.13}$$

The equality in (6.11) reflects the fact that the optimum detection ordering must maximize the total system capacity. Equality (6.12) follows from substituting (6.9) into (6.11) and equality (6.13) is because $\ln(x)$ is an increasing function of x. ■

Note that the optimum harmonic mean per-stream power gain

$$\overline{g^*} = \max_{\pi} \overline{g}(\pi) \tag{6.14}$$

depends only on **H**, not on SNR. Proposition 13 follows.

Proposition 13: In the coded V-BLAST under the IPA and capacity-achieving temporal codes at each stream, the optimum ordering depends only on **H** and it is SNR independent.

Two Tx antennas

When using two Tx antennas, the detection ordering under the IPA can be established based on the per-stream before-processing channel gains as stated in the following theorem.

Theorem 3: The optimum detection ordering for the coded V-BLAST with two Tx and n Rx antennas under the IPA is to detect the stream with higher before-detection channel gain first[10]:

$$\pi^* = \arg\max_{\pi} \overline{g}(\pi) = \{1,2\} \quad \text{iff} \quad |\mathbf{h}_1| > |\mathbf{h}_2| \quad \text{for } \forall\varphi \neq 0 \tag{6.15}$$

The "only if" part in (6.15) is true when $\varphi \neq \pi/2$. When $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ any ordering is optimum.

Proof: For the case of two Tx antennas, the channel matrix is given by $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2]$. To prove the "if" part, without loss of generality, it can be further assumed that $|\mathbf{h}_1| > |\mathbf{h}_2|$.

[10] this is the Foschini ordering.

There are two possible orderings: $\pi_1 = \{1, 2\}$ and $\pi_2 = \{2,1\}$. The harmonic mean per-stream power gains for these orderings are

$$\overline{g}(\pi_1) = \left[\frac{1}{2}\left(\frac{1}{|\mathbf{h}_{1\perp 2}|^2} + \frac{1}{|\mathbf{h}_2|^2}\right)\right]^{-1} = \frac{2|\mathbf{h}_{1\perp 2}|^2 |\mathbf{h}_2|^2}{|\mathbf{h}_{1\perp 2}|^2 + |\mathbf{h}_2|^2} \tag{6.16}$$

$$\overline{g}(\pi_2) = \left[\frac{1}{2}\left(\frac{1}{|\mathbf{h}_{2\perp 1}|^2} + \frac{1}{|\mathbf{h}_1|^2}\right)\right]^{-1} = \frac{2|\mathbf{h}_{2\perp 1}|^2 |\mathbf{h}_1|^2}{|\mathbf{h}_{2\perp 1}|^2 + |\mathbf{h}_1|^2} \tag{6.17}$$

where $\mathbf{h}_{i\perp j}$ refers to the projection of $\mathbf{h}_i$ orthogonal to $\mathbf{h}_j$. Equations (6.16) and (6.17) follow from evaluating (6.8) for the case of $m = 2$ under the orderings given by $\pi_1 = \{1, 2\}$ and $\pi_2 = \{2,1\}$ respectively. From Proposition 12, π_1 is optimum provided that

$$\overline{g}(\pi_1) > \overline{g}(\pi_2) \tag{6.18}$$

and the set of inequalities follows:

$$\frac{2|\mathbf{h}_1|^2 |\mathbf{h}_2|^2 \sin^2\varphi}{|\mathbf{h}_1|^2 \sin^2\varphi + |\mathbf{h}_2|^2} > \frac{2|\mathbf{h}_1|^2 |\mathbf{h}_2|^2 \sin^2\varphi}{|\mathbf{h}_2|^2 \sin^2\varphi + |\mathbf{h}_1|^2} \tag{6.19}$$

$$\Rightarrow |\mathbf{h}_2|^2 \sin^2\varphi + |\mathbf{h}_1|^2 > |\mathbf{h}_1|^2 \sin^2\varphi + |\mathbf{h}_2|^2 \tag{6.20}$$

$$\Rightarrow |\mathbf{h}_1|^2 \left(1 - \sin^2\varphi\right) > |\mathbf{h}_2|^2 \left(1 - \sin^2\varphi\right) \tag{6.21}$$

$$\Rightarrow |\mathbf{h}_1|^2 > |\mathbf{h}_2|^2 \tag{6.22}$$

Inequality (6.19) follows from substituting (6.16) and (6.17) into (6.18) and the use of the geometrical representation (4.14) i.e. $|\mathbf{h}_{i\perp j}| = |\mathbf{h}_i \sin\varphi|$ where φ is the angle between $\mathbf{h}_i$ and $\mathbf{h}_j$. Inequalities (6.20), (6.21) and (6.22) are the result of some straightforward mathematical manipulations. It is evident that the same chain of inequalities holds in the reverse direction i.e. starting in (6.22) and ending in (6.18), so the “only if” part is also proved and (6.15) follows. Note that if $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ the inequality (6.19) becomes an equality showing that any ordering is optimum. If $\varphi = 0$, the column vectors of the channel matrix are parallel. Therefore, the length of the projected vector is zero and thus the transmission strategy is to keep only one active stream (the stream with highest before-detection channel gain). ■

6.1.2. Suboptimum ordering

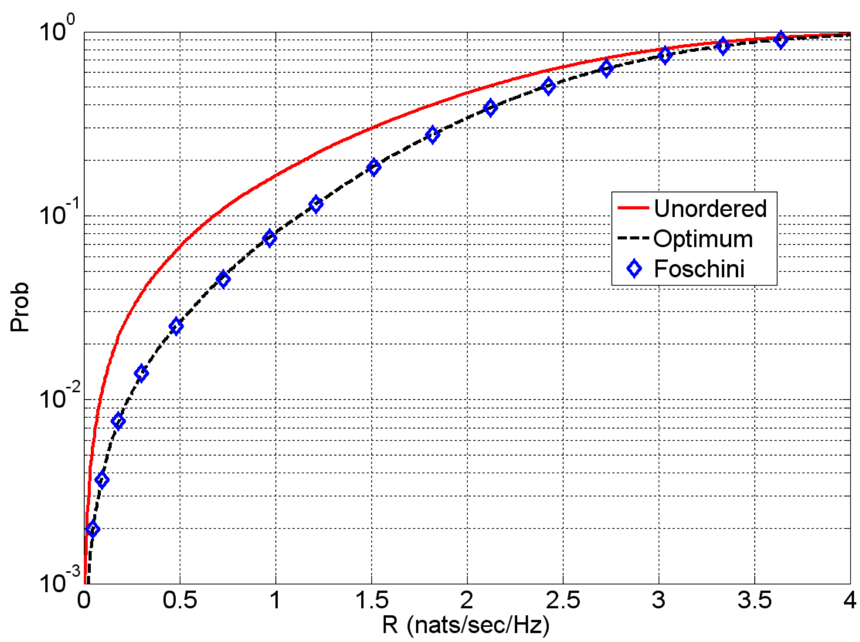

Figure 19: Empirical CDFs of the Foschini detection ordering, the optimum detection ordering and the unordered detection under the IPA for the (3x3) system; SNR=0dB; 10^5 channel realizations.

The optimality of the Foschini ordering for the coded V-BLAST under the URA was demonstrated in the previous chapter. Numerical simulations show that, although this ordering strategy is not optimum, in general, for the coded V-BLAST under the IPA when $m \geq 3$, it performs very close to the optimum one. Its performance is shown in Figures 19-20 where the empirical CDF graphs of the optimum detection ordering, the Foschini detection ordering and the unordered detection are illustrated for different values of m and SNR. In both cases, the simulations are based on 10^5 i.i.d Rayleigh fading channel realizations.

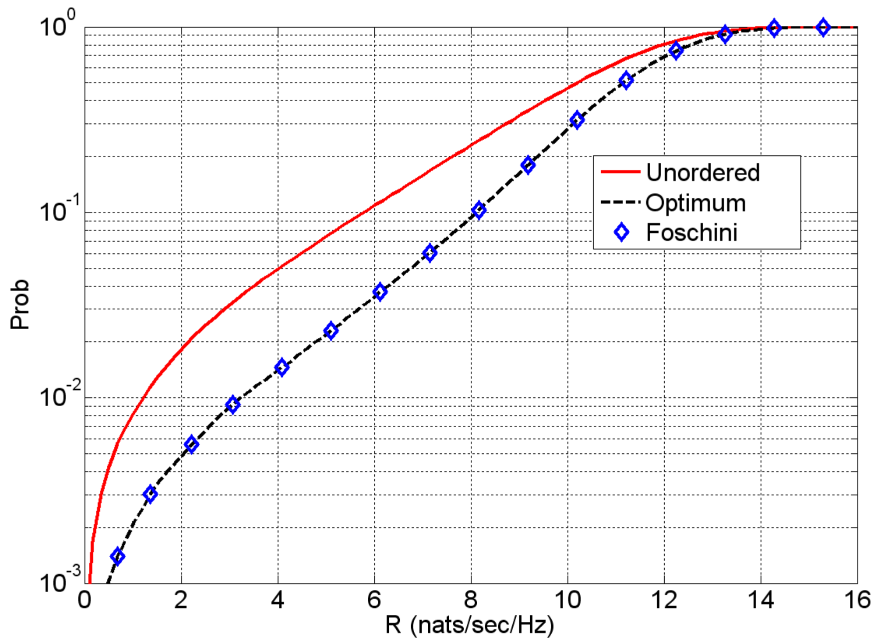

Figure 20: Empirical CDFs of the Foschini detection ordering, the optimum detection ordering and the unordered detection under IPA for the (4x4) system; SNR=10dB; 10^5 channel realizations.

6.2 Instantaneous Power and Rate Allocation (IPRA)

The optimum instantaneous power allocation strategy, subject to the total power constraint, when the per-stream rates match the per-stream capacities is discussed in this section. Based on the previous discussion, the per-streams rate under the IPRA is given by:

$$R_i = C_i = \ln\left(1+\alpha_i \left|\mathbf{h}_{i\perp}\right|^2 \gamma_0\right) \qquad \text{[nat/s/Hz]} \tag{6.23}$$

The total system capacity is

$$C = \sum_{i=1}^{m} C_i, \quad C_i = \ln\left(1+\alpha_i \left|\mathbf{h}_{i\perp}\right|^2 \gamma_0\right) \tag{6.24}$$

and the optimization problem for a given order π can be formulated as follows:

$$C_{IPRA}(\pi) = \max_{\boldsymbol{\alpha}(\pi)} \sum_i \ln\left(1 + \alpha_i(\pi)\left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right)$$
$$\text{s.t.} \quad \sum_i \alpha_i(\pi) = m, \quad \alpha_i(\pi) \geq 0 \tag{6.25}$$

It was demonstrated in [32] that the well-known waterfilling (WF) algorithm does not offer the optimal solution to this problem in general. The WF algorithm is optimum when the channel gains are independent of the allocated power. But the interference nulling step of the V-BLAST algorithm forces the channel gains to depend on the allocated power. If the power assigned to stream *i* is equal to zero, this stream is not active and it is not necessary to project out its interference from lower level streams. Therefore, turning off stream *i* affects the gain of lower level streams i.e. $\left|\mathbf{h}_{1\perp}\right|^2 \quad \ldots \quad \left|\mathbf{h}_{i-1\perp}\right|^2$. This results in a non-convex problem.

The solution to this problem is the fractional waterfilling algorithm (FWF) [32]. The original non-convex problem is split into 2^{m-1} convex sub-problems, one for each set of inactive transmitters, for which the solution is the WF algorithm. The optimum power allocation is given by the power allocation vector corresponding to the set with the highest capacity. The FWF algorithm can be summarized in three main steps:

1- Take 2^{m-1} sets of inactive streams (the first stream is always active because turning it off does not affect the rest of streams gains).

2- Apply WF to each set.

3- Take the power allocation vector corresponding to the set with the highest capacity.

The complete proof of the optimality of the FWF algorithm for the coded V-BLAST under the IPRA can be found in [32].

6.2.1. Optimum ordering under the IPRA

General case

In order to obtain the optimum ordering strategy based on the FWF approach it is necessary to repeat the three steps mentioned above $m!$ times (m is the number of Tx antennas) i.e. apply FWF to each possible ordering and select the ordering with the maximum capacity. This results in a very complex and lengthy process. However, extensive numerical simulations show the following:

Observation 4: The optimum ordering under the FWF is also optimum under the WF, and it offers the same value of the capacity[11]:

$$\pi^*_{WF} = \pi^*_{FWF} \tag{6.26}$$

$$C_{WF}\left(\pi^*_{WF}\right) = C_{FWF}\left(\pi^*_{FWF}\right) \tag{6.27}$$

From Observation 4, it can be argued that there is no loss of optimality when considering conventional WF to find the optimum ordering in the coded V-BLAST under the IPRA. This is formally stated in the following proposition:

Proposition 14: The optimum ordering in the coded V-BLAST under the IPRA with capacity achieving temporal codes at each stream is given by:

$$\pi^* = \arg\max_{\pi} \sum_i \ln\left(1+\alpha_i^*(\pi)\left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right) \tag{6.28}$$

where the optimum power allocation is given by the conventional WF algorithm i.e.

$$\alpha_i^*(\pi) = \left[\mu(\pi) - \frac{1}{\gamma_0 \left|\mathbf{h}_{i\perp}(\pi)\right|^2}\right]_+ \tag{6.29}$$

[11] the analytical proof is not available.

where $\mu(\pi)$ is the water level for a given order π. $\mu(\pi)$ is calculated from the total power constraint. More information about the WF algorithm can be found in Appendix A.

Two Tx antennas

Note: The derivations in the following section make use of the results for the WF algorithm derived in Appendix A.

Under the IPRA when the channel gain of a given stream is too weak to be transmitted, that stream is turned off by assigning zero power to it. The problem of finding the optimum ordering under the IPRA for the case of two Tx antennas can be divided into three sub-problems based on three SNR regimes.

- ***Low SNR regime:*** Both orderings have one active stream
- ***Intermediate SNR regime:*** The optimum ordering has one active stream and the suboptimum has two active streams.
- ***High SNR regime:*** Both orderings have two active streams.

The case where the suboptimum ordering has one active stream and the optimum has two active streams never takes place. This follows because, from the WF algorithm, the SNR threshold at which the suboptimum ordering must start operating with two active streams is lower than the SNR threshold corresponding to the optimum ordering. This statement is analytically proved in the next section.

Working with the capacity equations for each ordering at each SNR regime independently, the detection ordering can be established based on the per-stream before-processing channel gains as stated in the following theorem.

Theorem 4: The optimum detection ordering for the coded V-BLAST with two Tx and n Rx antennas under the IPRA is to detect the stream with highest before-detection channel gain at the last step:

$$\pi^* = \arg\max_{\pi} \sum_{i=1}^{2} \ln\left(1+\alpha_i^*(\pi)\left|\mathbf{h}_{i\perp}(\pi)\right|^2 \gamma_0\right) = \{1,2\} \quad \text{iff} \quad |\mathbf{h}_1| < |\mathbf{h}_2| \tag{6.30}$$

The "only if" part in (6.30) is true when $\varphi \neq \pi/2$. When $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ any ordering is optimum.

Proof: See Appendix B.

We are now ready to compare the optimal ordering strategies for the coded V-BLAST under the IRA and the IPRA.

Proposition 15: The optimal orderings in the coded V-BLAST with two Tx antennas under the IRA and under the IPRA are the same[12], i.e. the stream with highest before detection channel gain is detected last.

Observation 5: The optimal orderings in the coded V-BLAST with m Tx antennas $(m > 2)$ under the IRA and under the IPRA[13] are not, in general, the same.

Proposition 15 is a direct consequence of Theorems 1 and 4 while Observation 5 is supported by extensive numerical simulations.

Some numerical simulations supporting Observation 5

The results of some numerical simulations supporting Observation 5 are described below. The optimal detection orderings for the $(3\text{x}3)$ coded V-BLAST under the IRA and the WF are shown for three Rayleigh fading channel realizations at different SNR values

[12] and the opposite of Foschini ordering.

[13] via WF.

(-10dB, 0dB and 10dB). The capacities of these orderings were calculated following both optimization strategies and are illustrated in the tables. The examples clearly show that there are cases where the optimum detection ordering under the IRA is suboptimum under the WF and vice versa.

The channel realization analysed at $\text{SNR} = 10\text{dB}$ is

$$\mathbf{H} = \begin{bmatrix} -0.1667 - 0.8681i & 0.6585 - 0.4900i & -1.0183 + 0.1238i \\ -0.3590 + 0.2313i & 0.2423 + 0.2329i & 0.3397 + 0.0829i \\ -0.5827 - 0.4670i & -0.4569 - 0.9415i & -0.5628 + 1.4247i \end{bmatrix}$$

Table 3: Performances of the optimal orderings under the IRA and the WF for both optimization strategies for the given channel realization at SNR = 10dB.

	$C_{IRA}\,(\text{nats/sec/Hz})$	$C_{WF}\,(\text{nats/sec/Hz})$
$\pi^*_{IRA} = \{1,2,3\}$	5.54	5.67
$\pi^*_{WF} = \{3,1,2\}$	5.51	5.72

The channel realization analysed at $\text{SNR} = 0\text{dB}$ is

$$\mathbf{H} = \begin{bmatrix} -0.0668 + 0.8285i & -0.2184 + 0.5226i & -0.1904 - 0.6955i \\ -0.1620 + 0.9091i & 0.9664 + 0.3449i & -1.4900 - 0.1004i \\ -0.4010 + 0.2689i & -1.0233 + 0.2497i & -0.0372 - 0.5228i \end{bmatrix}$$

Table 4: Performances of the optimal orderings under the IRA and the WF for both optimization strategies for the given channel realization at SNR = 0dB.

	$C_{IRA}\,(\text{nats/sec/Hz})$	$C_{WF}\,(\text{nats/sec/Hz})$
$\pi^*_{IRA} = \{1,2,3\}$	2.40	2.56
$\pi^*_{WF} = \{3,1,2\}$	2.38	2.57

The channel realization analysed at $\mathrm{SNR} = -10\mathrm{dB}$ is

$$\mathbf{H} = \begin{bmatrix} 0.3171-0.1795i & -0.4877+0.2142i & -1.0822+0.3842i \\ -1.5697-0.7095i & -1.0072+0.4643i & 1.1884-0.5056i \\ 0.0550-0.4633i & -0.5429-0.1710i & 0.2563-0.0850i \end{bmatrix}$$

Table 5: Performances of the optimal orderings under the IRA and the WF for both optimization strategies for the given channel realization at $\mathrm{SNR} = -10\mathrm{dB}$.

	$C_{IRA}\left(\text{nats/sec/Hz}\right)$	$C_{WF}\left(\text{nats/sec/Hz}\right)$
$\pi^*_{IRA} = \{1,2,3\}$	0.42	0.65
$\pi^*_{WF} = \{3,2,1\}$	0.36	0.69

6.2.2. Suboptimum ordering

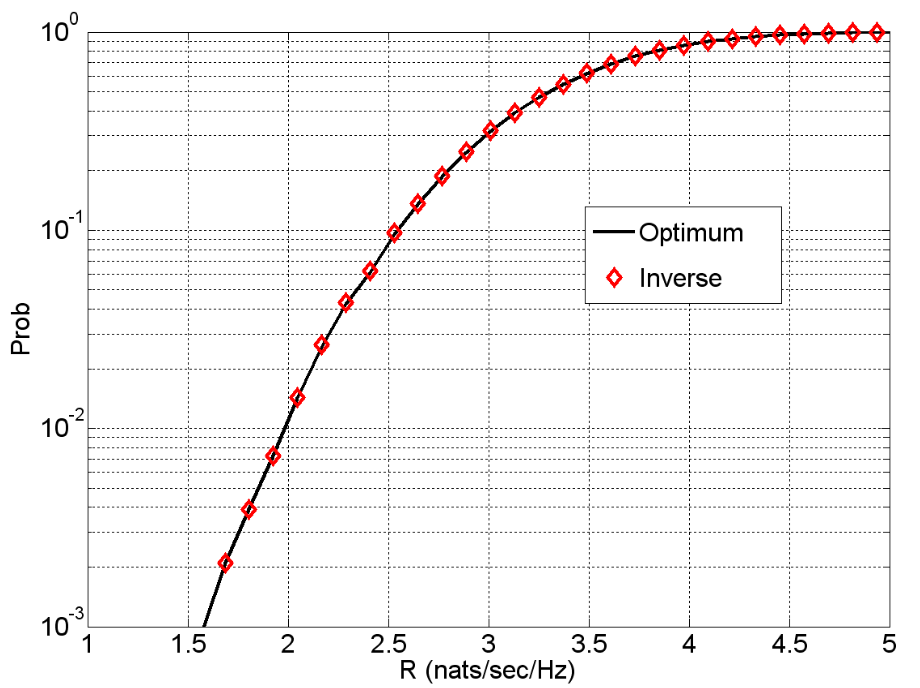

Figure 21: Empirical CDFs of the inverse ordering and the optimum ordering under WF for the (3x3) system; SNR=0dB; 10^4 channel realizations.

In section 4.2, it was shown that the inverse ordering performs very close to the optimum one for the coded V-BLAST under the IRA. In this section its performance is evaluated under the WF, the FWF and compared with the optimum ordering under IPRA using simulations.

Figures 21-23 show the empirical CDF curves, based on 10^4 i.i.d. Rayleigh fading channel realizations, of the Inverse detection ordering and the optimum one for the $(3x3)$ and $(4x4)$ coded systems respectively under the IPRA at different values of SNR. The WF power allocation is used in both orderings. The almost optimum performance of the inverse ordering is evident.

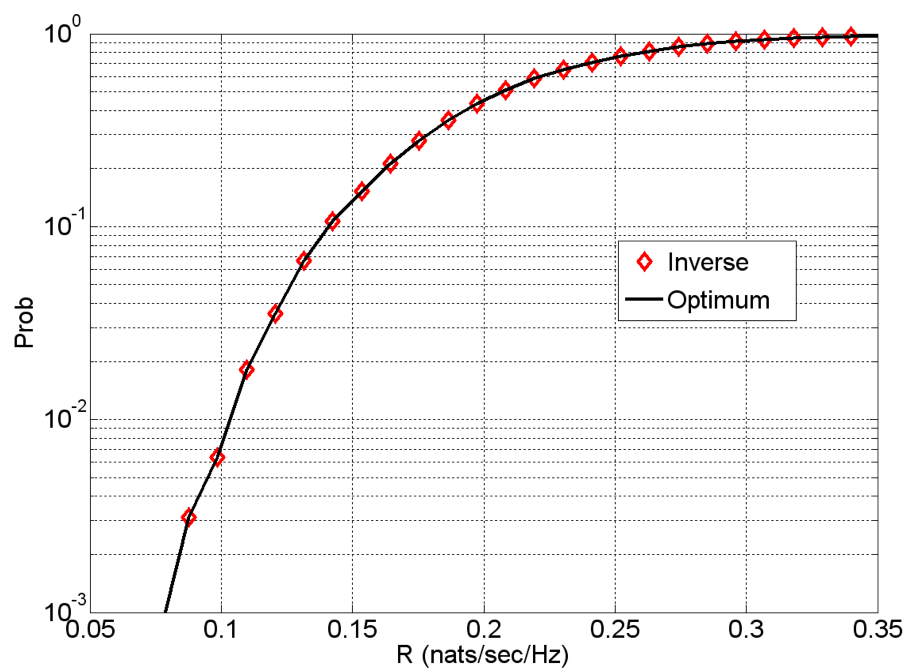

Figure 22: Empirical CDFs of the inverse ordering and the optimum ordering under WF for the (4x4) system; SNR=-20dB; 10^4 channel realizations.

Please note that the WF is not, in general, the optimum power/rate allocation for the inverse ordering (Observation 4 is only valid for the optimum ordering), the optimum power/rate allocation for the inverse ordering is the FWF. However, this ordering offers the same value of capacity under the WF and the FWF with very high probability. To show this, the capacity of the inverse ordering was also evaluated under the FWF (for the same channel realizations and parameters used in Figures 21, 22 and 23) and the results are illustrated in Table 6.

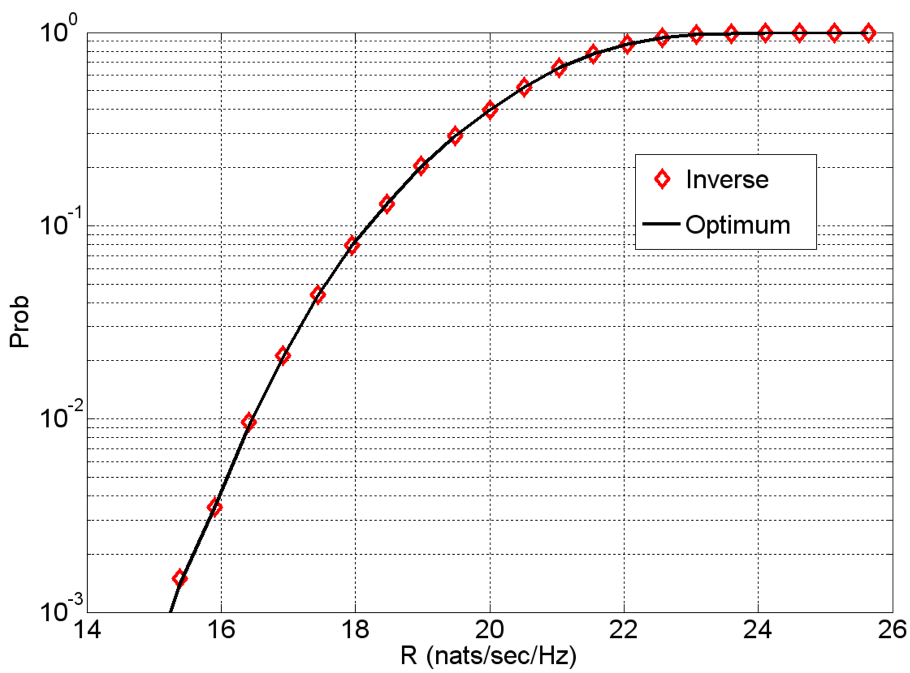

Figure 23: Empirical CDFs of the inverse ordering and the optimum ordering under WF for the (4x4) system; SNR=20dB; 10^4 channel realizations.

Table 6: The inverse ordering achieving the same capacity under the WF and the FWF with high probability.

	The same parameters used in Figure 21	The same parameters used in Figure 22	The same parameters used in Figure 23
$\Pr(C_{WF} = C_{FWF})$	0.98	0.999	0.998
$\Pr(C_{WF} \neq C_{FWF})$	0.02	10^{-3}	$2x10^{-3}$

6.2.3. Analytical boundaries of the SNR regimes for two Tx antennas

The case of two Tx antennas under the IPRA can be studied in three different SNR regimes: low SNR, intermediate SNR and high SNR. The objective of this section is to derive the analytical expressions of the SNR thresholds defining the boundaries of each of these regimes.

Low SNR regime

In the low SNR regime, we have the two possible orderings operating with only one active stream, i.e. the optimum IPRA strategy to maximize the transmission rate is to transmit with one antenna in both cases. It is a property of the WF that, for a given ordering, only one stream is active at low SNR (see Appendix A). Therefore, as long as the SNR is sufficiently low both orderings will have only one active stream. The SNR threshold defining this regime is shown in the following proposition:

Proposition 16: The low SNR regime includes all SNR values satisfying

$$\gamma_0 \leq \frac{1}{2}\left|\frac{1}{g_1} - \frac{1}{g_2\beta}\right| \tag{6.31}$$

where $g_i = |\mathbf{h}_i|^2$ and $\sin^2\varphi = \beta$.

Proof: For the case of two Tx antennas, the two possible orderings are: $\pi_1 = \{1,2\}$ and $\pi_2 = \{2,1\}$. Without loss of generality, it can be further assumed that $g_1 < g_2$. Then, from Theorem 4, π_1 is the optimum ordering. This ordering will operate with only one active stream provided that

$$\mu_1 \leq \frac{1}{\min\left[g_1\beta\gamma_0, g_2\gamma_0\right]} = \frac{1}{g_1\beta\gamma_0} \tag{6.32}$$

The inequality (6.32) follows from the per-stream power allocation formula in the WF algorithm (see Appendix A) and the equality follows from $g_1 < g_2$ and the fact that the projection of a vector cannot increase its norm i.e. $\beta \leq 1$. μ_1 is the water level corresponding to π_1 given by

$$\mu_1 = 1 + \frac{1}{2\gamma_0}\left(\frac{1}{g_1\beta} + \frac{1}{g_2}\right) \tag{6.33}$$

The complete derivation to obtain (6.33) can be found in Appendix A. Substituting (6.33) into (6.32), the condition under which π_1 will operate with only one active stream can be written as

$$1 + \frac{1}{2\gamma_0}\left(\frac{1}{g_1\beta} + \frac{1}{g_2}\right) \leq \frac{1}{g_1\beta\gamma_0} \tag{6.34}$$

and after some mathematical manipulations, (6.34) can be expressed as

$$\gamma_0 \leq \frac{g_2 - g_1\beta}{2g_1g_2\beta} \tag{6.35}$$

Ordering π_1 will operate with only one active stream for all SNR values satisfying (6.35).

The same reasoning applies to the second ordering: Based on the WF per-stream power allocation formula, π_2 will operate with only one active stream provided that

$$\mu_2 \leq \frac{1}{\min\left[g_2\beta\gamma_0, g_1\gamma_0\right]} \tag{6.36}$$

where μ_2 is the corresponding water level given by

$$\mu_2 = 1 + \frac{1}{2\gamma_0}\left(\frac{1}{g_2\beta} + \frac{1}{g_1}\right) \tag{6.37}$$

Substituting (6.37) into (6.36) the condition under which π_2 will operate with only one active stream can be written as

$$1 + \frac{1}{2\gamma_0}\left(\frac{1}{g_2\beta} + \frac{1}{g_1}\right) \leq \frac{1}{\min\left[g_2\beta\gamma_0, g_1\gamma_0\right]} \tag{6.38}$$

and two possible scenarios arise. The first scenario takes place when

$$\min\left[g_2\beta\gamma_0, g_1\gamma_0\right] = g_2\beta\gamma_0 \tag{6.39}$$

and equation (6.38) reduces to

$$\gamma_0 \le \frac{g_1 - g_2\beta}{2g_1g_2\beta} \tag{6.40}$$

Equation (6.40) follows after substituting (6.39) into (6.38). The second scenario takes place when

$$\min\left[g_2\beta\gamma_0, g_1\gamma_0\right] = g_1\gamma_0 \tag{6.41}$$

and equation (6.38) reduces to

$$\gamma_0 \le \frac{g_2\beta - g_1}{2g_1g_2\beta} \tag{6.42}$$

Equation (6.42) follows after substituting (6.41) into (6.38). From (6.40) and (6.42), ordering π_2 will operate with only one active stream for all SNR values satisfying:

$$\gamma_0 \le \frac{|g_1 - g_2\beta|}{2g_1g_2\beta} \tag{6.43}$$

Based on the fact that

$$g_2 - g_1\beta > |g_1 - g_2\beta| \quad \text{if} \quad g_2 > g_1 \tag{6.44}$$

the SNR thresholds in (6.35) and (6.43) can be compared as[14]

$$\frac{|g_1 - g_2\beta|}{2g_1g_2\beta} < \frac{g_2 - g_1\beta}{2g_1g_2\beta} \tag{6.45}$$

therefore, both orderings will operate with one active stream if

$$\gamma_0 \le \frac{|g_1 - g_2\beta|}{2g_1g_2\beta} \tag{6.46}$$

And (6.31) follows. ■

[14] Inequality (6.45) also shows that the case where the suboptimum ordering has one active stream and the optimum has two active streams never takes place.

High SNR regime

In the high SNR regime, both orderings operate with two active streams, i.e. the optimum IPRA strategy to maximize the transmission rate is to transmit with two antennas in both cases. From the WF algorithm, if the received SNR is high enough in both orderings, they will operate with two active streams. The SNR threshold defining this regime is shown in the following proposition.

Proposition 17: The high SNR regime includes all SNR values satisfying

$$\gamma_0 > \frac{1}{2}\left(\frac{1}{g_1\beta} - \frac{1}{g_2}\right) \tag{6.47}$$

Proof: This proof is based on the results obtained for the low SNR regime case. From (6.35), the optimum ordering will operate with two active streams when

$$\gamma_0 > \frac{g_2 - g_1\beta}{2g_1g_2\beta} \tag{6.48}$$

From (6.43) the suboptimum ordering will operate with two active streams when

$$\gamma_0 > \frac{|g_1 - g_2\beta|}{2g_1g_2\beta} \tag{6.49}$$

Then both orderings will operate with two active streams provided that

$$\gamma_0 > \frac{g_2 - g_1\beta}{2g_1g_2\beta} \tag{6.50}$$

Equation (6.50) is a direct consequence of (6.45) and equation (6.47) follows. ■

Intermediate SNR regime

In the intermediate SNR regime, the optimum ordering operates with only one active stream while the suboptimum ordering operates with both streams active, i.e. in the case of π_1 the optimum IPRA strategy to maximize the transmission rate is to transmit with only one antenna while in the case of π_2 the optimum IPRA strategy is to transmit with both antennas. The SNR thresholds defining this regime are shown in the following proposition.

Proposition 18: The intermediate SNR regime includes all SNR values satisfying

$$\frac{1}{2}\left|\frac{1}{g_1}-\frac{1}{g_2\beta}\right| < \gamma_0 \leq \frac{1}{2}\left(\frac{1}{g_1\beta}-\frac{1}{g_2}\right) \qquad (6.51)$$

Proof: Equation (6.51) follows from Propositions 16 and 17. ■

Please note that when $g_1 = g_2$ and/or $\beta = 1$, the intermediate SNR regime does not exist. This can be easily verified by substituting these values into (6.51).

SNR regimes: an example

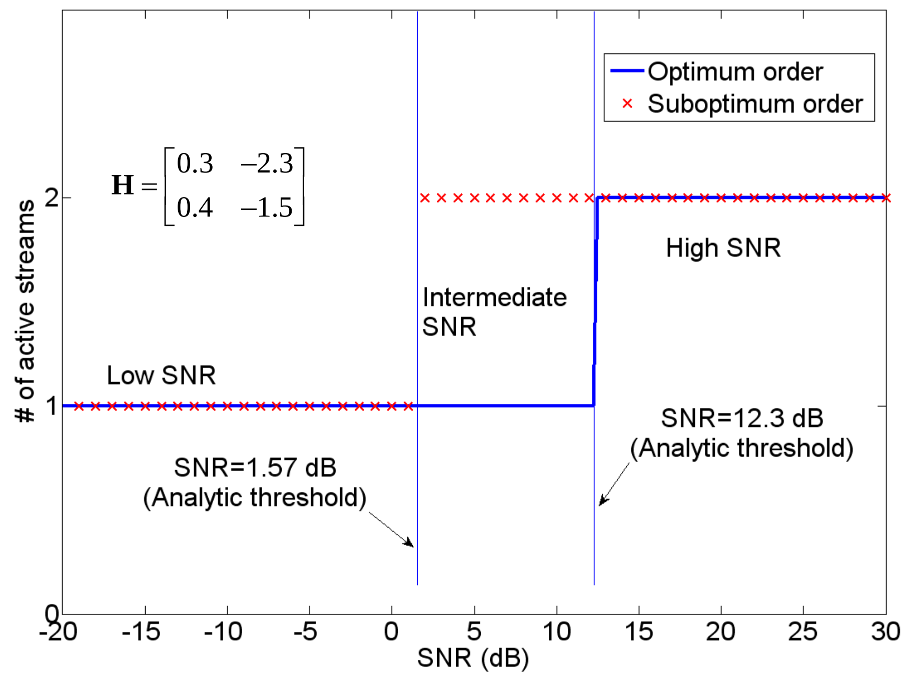

Figure 24: Number of active streams vs. SNR for a given channel realization for the (2x2) system under the IPRA.

Numerical simulations were undertaken to validate the analytic results of the different SNR regimes thresholds. Figure 24 plots the number of active streams as a function of SNR of the $(2x2)$ coded V-BLAST system under the IPRA for a given Rayleigh fading channel realization. The performances of both the optimum and the suboptimum ordering are illustrated. It can be seen that the analytical thresholds of the SNR regimes agree with the numerical ones. The behaviour of the number of active streams is described as follows: as long as SNR is low enough, both orderings start transmitting with only one antenna (low SNR regime), at $\mathrm{SNR} = 1.6\mathrm{dB}$ the system with the suboptimal ordering switches to two active streams while the system with the optimum ordering is transmitting with only one antenna (Intermediate SNR regime). Finally, at $\mathrm{SNR} = 12.3\mathrm{dB}$ both orderings transmit with two streams (High SNR regime).

6.3. Summary

The optimum detection orderings for the coded V-BLAST under the IPA and the IPRA were studied.

The optimum detection ordering under the IPA is SNR independent and it maximizes the harmonic mean per-stream power gain. For the case of two Tx antennas, the optimum ordering can be established based on the before-processing channel gains: the stream with highest before-processing channel gain must be detected first. It was shown that the Foschini ordering, although suboptimum, offers an outstanding performance for $m \geq 3$.

There is no loss of optimality when considering conventional WF to find the optimum ordering in the coded V-BLAST under the IPRA. This is because the optimum ordering under FWF is exactly the same as the optimum ordering under WF, and it achieves the same value of capacity. For the 2-Tx-antenna case, the optimum ordering can be established based on the before-processing channel gains: the stream with highest before-processing channel gain must be detected last. Analytical SNR thresholds, which separate the low, intermediate and high SNR regimes, were derived. It was shown that the inverse ordering under the WF performs almost optimally.

7. SNR gain of ordering

The SNR gain of ordering under the IPRA is discussed in this chapter. To explore the properties of this gain for two Tx antennas, we rely only on analytical techniques. However, for $m > 2$, numerical simulations are used due to the complexity of the mathematical analysis.

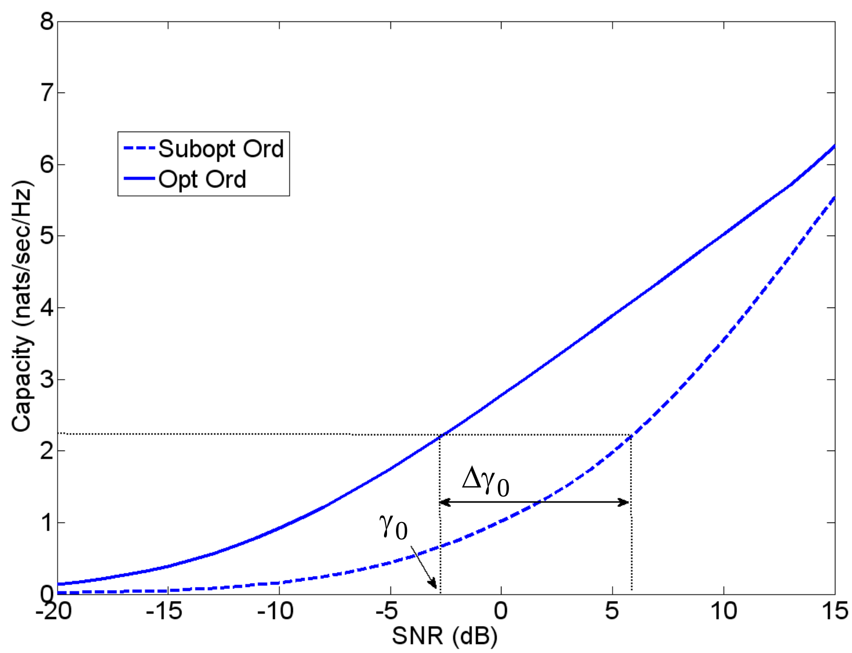

Figure 25: Definition of the SNR gain of ordering.

Recall that in the coded V-BLAST the optimization of capacity and outage probability achieve the same lowest value of outage probability. Based on this fact, the SNR gain of ordering in the coded V-BLAST is defined as the difference in SNR required by the unordered V-BLAST to achieve the same capacity as the optimally ordered i.e.

$$C_{\pi^*}(\gamma_0) = C_{\pi}(G\gamma_0) \tag{7.1}$$

where the left-hand side represents the system using the optimum detection ordering and the right-hand side represents the system using the unordered detection. G is the SNR gain of the optimum ordering procedure. Figure 25 is an illustrative example of the SNR gain definition.

7.1. Two Tx antennas

The main objective of this section is to derive closed-form expressions for the SNR gain of ordering in the 2-Tx-antenna coded V-BLAST under the IPRA with capacity achieving temporal codes. Some propositions follow from the analytical results providing a deep insight into the SNR gain offered by the optimum detection ordering procedure.

As in previous sections, the analysis for two Tx antennas will be divided into three SNR regimes: low, intermediate and high. Analytical expressions for the SNR gain of ordering at each regime are given below.

Low SNR regime: $\gamma_0 \le \frac{1}{2}\left|\frac{1}{g_1} - \frac{1}{g_2\beta}\right|$

Proposition 19: The SNR gain of the optimum ordering procedure in the low SNR regime is given by:

$$G = \min\left[\frac{1}{\beta}, \frac{g_2}{g_1}\right] \tag{7.2}$$

where $g_i = |\mathbf{h}_i|^2$ and $\sin^2\varphi = \beta$.

Proof: For the case of two Tx antennas, the channel matrix is given by $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2]$. The two possible orderings are: $\pi_1 = \{1,2\}$ and $\pi_2 = \{2,1\}$. Without loss of generality, it can be further assumed that $|\mathbf{h}_1| < |\mathbf{h}_2|$. Then, from Theorem 4, $\pi_1 = \{1,2\} = \pi^*$ is the optimum ordering, i.e. it maximizes the system capacity.

The capacity equations for orderings π_1 and π_2 in the low SNR regime for a given channel realization are

$$C_{WF}\left(\pi_1\right) = \ln\left(1+2g_2\gamma_0\right) \tag{7.3}$$

$$C_{WF}\left(\pi_2\right) = \ln\left(1+2\max\left[g_2\beta, g_1\right]\gamma_0\right) \tag{7.4}$$

Equations (7.3) and (7.4) follow from making the appropriate substitutions of the channel gains in (11.11) depending on the specific ordering. For (7.3), it has been also used the fact that $g_1 < g_2$ and $\beta \le 1$. From the definition of the SNR gain, the following set of equalities holds:

$$C_{\pi_1}\left(\gamma_0\right) = C_{\pi_2}\left(G\gamma_0\right) \tag{7.5}$$

$$\Rightarrow \ln\left(1+2g_2\gamma_0\right) = \ln\left(1+2\max\left[g_2\beta, g_1\right]G\gamma_0\right) \tag{7.6}$$

$$\Rightarrow G = \frac{g_2}{\max\left[g_2\beta, g_1\right]} \tag{7.7}$$

Equality (7.5) states the formal definition of the SNR gain of ordering based on π_1 and π_2. Equality (7.6) results from substituting (7.3) and (7.4) into (7.5). Equality (7.7) follows after simple mathematical manipulations applied to (7.6). Finally, (7.2) follows from (7.7). ■

Some conclusions arise from (7.2) and are stated in the following proposition.

Proposition 20: In the low SNR regime the SNR gain of ordering has the following properties:

- It is SNR independent, i.e. it is constant for all low SNR values.
- It depends on the ratio g_2/g_1 or $1/\beta$.
- If $g_1 = g_2$ and/or $\beta = 1$ there is no gain, i.e. if the channel gains are equal and/or the channel columns are orthogonal to each other, then the SNR gain of ordering is equal to one (both orderings offer the same capacity).

Note: The last property holds for all the SNR regimes; hence it will not be repeated again.

High SNR regime: $\gamma_0 > \frac{1}{2}\left(\frac{1}{g_1\beta} - \frac{1}{g_2}\right)$

Proposition 21: The SNR gain of the optimum ordering procedure in the high SNR regime is expressed as:

$$G = 1 + \frac{(1-\beta)(g_2 - g_1)}{2 g_1 g_2 \beta \gamma_0} \tag{7.8}$$

Proof: The capacity equations for orderings π_1 and π_2 in the high SNR regime for a given channel realization can be expressed as

$$C_{WF}(\pi_1) = \ln\left(1 + \alpha_1^* g_1 \beta \gamma_0\right) + \ln\left(1 + \alpha_2^* g_2 \gamma_0\right) \tag{7.9}$$

$$C_{WF}(\pi_2) = \ln\left(1 + \alpha_a^* g_2 \beta \gamma_0\right) + \ln\left(1 + \alpha_b^* g_1 \gamma_0\right) \tag{7.10}$$

Equations (7.9) and (7.10) follow from making the appropriate substitutions of the channel gains in (11.4) depending on the specific ordering. The pairs α_1^*, α_2^* and α_a^*, α_b^* are the optimum waterfilling power allocations for the orderings π_1 and π_2 respectively. Equations (7.9) and (7.10) can be expressed in terms of the corresponding water levels as:

$$C_{WF}(\pi_1) = \ln\left(\mu_1^2 g_1 \beta g_2 \gamma_0\right) \tag{7.11}$$

$$C_{WF}(\pi_2) = \ln\left(\mu_2^2 g_2 \beta g_1 \gamma_0\right) \tag{7.12}$$

This follows from making the appropriate substitutions of the channel gains and the water levels in (11.7) depending on the specific ordering. The water levels μ_1 and μ_2 are as in (6.33) and (6.37) respectively. Substituting (6.33) and (6.37) into (7.11) and (7.12) respectively, the capacity equations can be expressed as:

$$C_{WF}(\pi_1) = \ln\left[\frac{(A\gamma_0 + B)^2}{2A}\right] \quad (7.13)$$

$$C_{WF}(\pi_2) = \ln\left[\frac{(A\gamma_0 + C)^2}{2A}\right] \quad (7.14)$$

where A, B and C are given by: $A = 2g_1g_2\beta$, $B = g_1\beta + g_2$, $C = g_2\beta + g_1$. From the definition of the SNR gain, the following set of equalities holds:

$$C_{\pi_1}(\gamma_0) = C_{\pi_2}(G\gamma_0) \quad (7.15)$$

$$\Rightarrow \ln\left[\frac{(A\gamma_0 + B)^2}{2A}\right] = \ln\left[\frac{(AG\gamma_0 + C)^2}{2A}\right] \quad (7.16)$$

$$\Rightarrow G = \frac{A\gamma_0 + B - C}{A\gamma_0} \quad (7.17)$$

Equality (7.15) states the formal definition of the SNR gain of ordering based on π_1 and π_2. Equality (7.16) results from substituting (7.13) and (7.14) into (7.15). Equality (7.17) follows after simple mathematical manipulations applied to (7.16). Finally, (7.8) follows from substituting the original expressions for A, B and C into (7.17). ■

Some conclusions arise from (7.8) and are stated in the following proposition.

Proposition 22: In the high SNR regime the SNR gain of ordering has the following properties:

- For given g_1, g_2 and β it is a decreasing function of SNR, i.e. $G \downarrow \gamma_0$
- For given g_1, g_2 and γ_0 it is a decreasing function of β, i.e. $G \downarrow \beta$
- For given g_2, β and γ_0 it is a decreasing function of g_1, i.e. $G \downarrow g_1$

Intermediate SNR regime: $\frac{1}{2}\left|\frac{1}{g_1}-\frac{1}{g_2\beta}\right| < \gamma_0 \le \frac{1}{2}\left(\frac{1}{g_1\beta}-\frac{1}{g_2}\right)$

Proposition 23: The SNR gain of the optimum ordering procedure in the intermediate SNR regime is expressed as:

$$G = \frac{1}{\gamma_0}\left(\sqrt{\frac{1+2g_2\gamma_0}{g_1 g_2 \beta}} - \frac{g_2\beta + g_1}{2 g_1 g_2 \beta}\right) \tag{7.18}$$

Proof: The capacity equations for orderings π_1 and π_2 in the intermediate SNR regime for a given channel realization are given by (7.3) and (7.10) respectively, i.e.

$$C_{WF}\left(\pi_1\right) = \ln\left(1+2g_2\gamma_0\right) \tag{7.19}$$

$$C_{WF}\left(\pi_2\right) = \ln\left(1+\alpha_a^* g_2 \beta \gamma_0\right) + \ln\left(1+\alpha_b^* g_1 \gamma_0\right) \tag{7.20}$$

Equations (7.19) and (7.20) follow because in the intermediate SNR regime the optimum ordering $\left(\pi_1\right)$ have only one active stream while the sub-optimum has both streams active. It was shown in the previous section that equation (7.20) can also be expressed as in (7.14). From the definition of the SNR gain, the following set of equations holds:

$$C_{\pi_1}\left(\gamma_0\right) = C_{\pi_2}\left(G\gamma_0\right) \tag{7.21}$$

$$\Rightarrow \ln\left(1+2g_2\gamma_0\right) = \ln\left[\frac{\left(AG\gamma_0 + C\right)^2}{2A}\right] \tag{7.22}$$

$$\Rightarrow G = \frac{\sqrt{2A\left(1+2g_2\gamma_0\right)} - C}{A\gamma_0} \tag{7.23}$$

Equality (7.21) states the formal definition of the SNR gain of ordering based on π_1 and π_2. Equality (7.22) results from substituting (7.19) and (7.14) into (7.21). Equality (7.23) follows after simple mathematical manipulations applied to (7.22). Finally, (7.18) follows from substituting the original expressions for A and C into (7.23). ■

Note that the SNR gain expression is more complicated than in the previous regimes. Proposition 24 states a conclusion that arise from (7.18):

Proposition 24: In the intermediate SNR regime the SNR gain of ordering is SNR dependent.

Conjecture: Based on previous properties of the SNR gain, we conjecture that G is bounded as,

$$1 \le G \le \min\left[\frac{1}{\beta}, \frac{g_2}{g_1}\right] \tag{7.24}$$

Note: This conjecture is suggested by the low and high SNR regimes. However, since $G(\gamma)$ may exhibit a non-monotonic behaviour in the intermediate SNR regime, we do not have a complete proof of this result at the moment.

7.1.1. Two Tx antennas: validation of the results

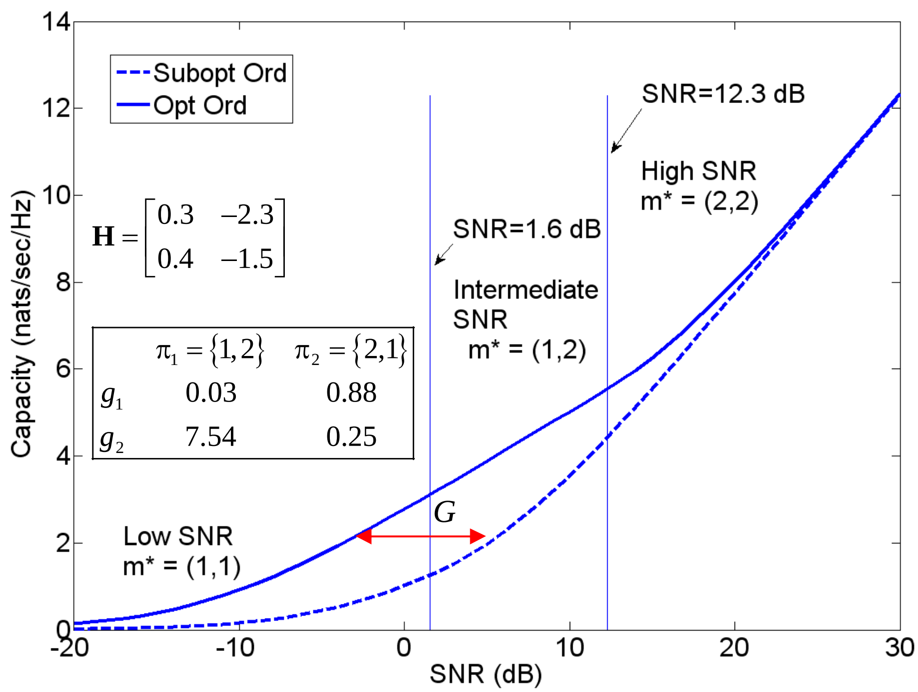

Figure 26: SNR gain of ordering for the (2x2) system.

Numerical simulations have been carried out to validate the closed-form expressions for the SNR gain of ordering at each SNR regime. Figure 26 plots the capacity as a function of SNR for the $(2x2)$ coded V-BLAST system under the IPRA (via WF) for a given Rayleigh fading channel realization. The gap in SNR for the same value of capacity is the SNR gain offered for the optimum ordering. The vertical lines correspond to the analytical expressions (6.31) and (6.47) and they separate the three SNR regimes.

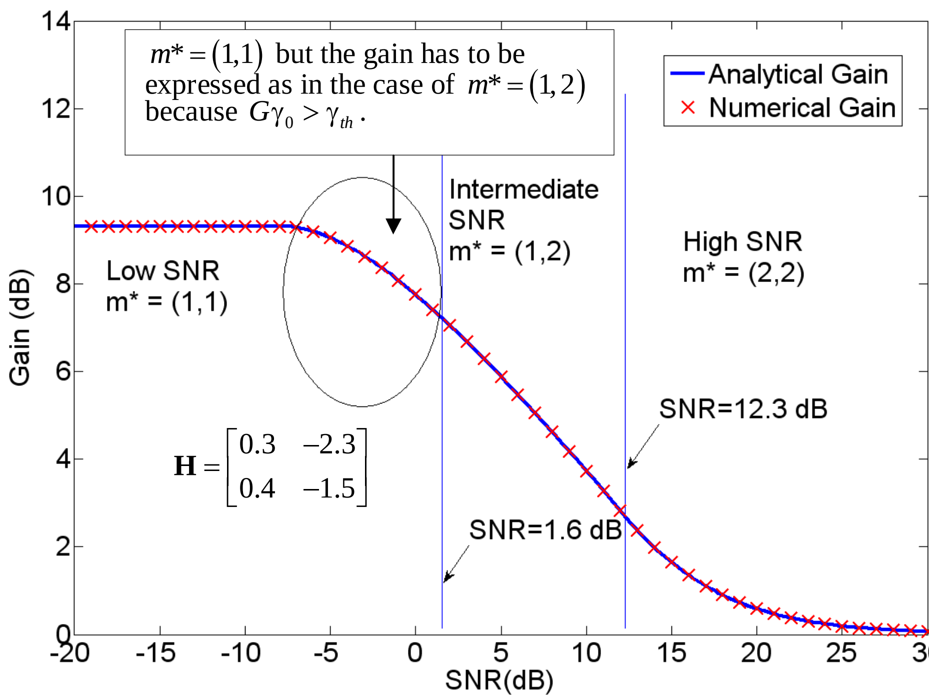

Figure 27: SNR gain vs. SNR; numerical and analytical solutions.

Figure 27 shows the SNR gain of ordering (numerical and analytical) as a function of SNR for the example shown in Figure 26. This graph helps us understand better the behaviour of the SNR gain. It can be seen how the numerical and analytical results perfectly match with each other, validating the analytical results. In the low SNR regime, the SNR gain is constant (independent of the SNR) when operate with $m^{*}=(1,1)$[15]. In the intermediate SNR regime, the SNR gain is still considerably high and, in this case, it is a decreasing function of the SNR. In the high SNR regime the SNR gain decreases as the SNR increases showing that the optimal ordering does not offer a significant advantage in this SNR regime.

7.2. Three transmit antennas: an example

The closed-form analysis when the number of Tx antennas is greater than two is very difficult. Therefore, we rely on numerical simulations to study the benefits of the optimum ordering procedure and the behaviour of the SNR gain of ordering in these systems. An example is discussed below.

Figure 28 illustrates the behaviour of the capacity as a function of SNR for the $(3\text{x}3)$ coded V-BLAST system under the WF for a given Rayleigh fading channel realization. Only four orderings from which some conclusions can be derived are shown[16]. Some observations follow from the graph:

- There is one ordering (Ordering D) that offers the worst performance at all SNR values and there is a considerable SNR gap between this ordering and the rest. Therefore, the gain provided by the optimum ordering procedure is evident.
- There are different orderings that offer the same optimum capacity, for example orderings A and C are optimum for $-10\text{dB} \le \gamma_0 \le 3\text{dB}$.

[15] Each ordering operates with one active stream.

[16] Two orderings that do not provide useful information are not shown to avoid overloading the graph.

- The optimal ordering is, in general, SNR dependent, i.e. there is not, in general, one ordering that remains optimum for all SNR values. For example orderings A and C are optimum for $-10\text{dB} \le \gamma_0 \le 3\text{dB}$ while ordering B is optimum for $\gamma_0 > 3\text{dB}$.

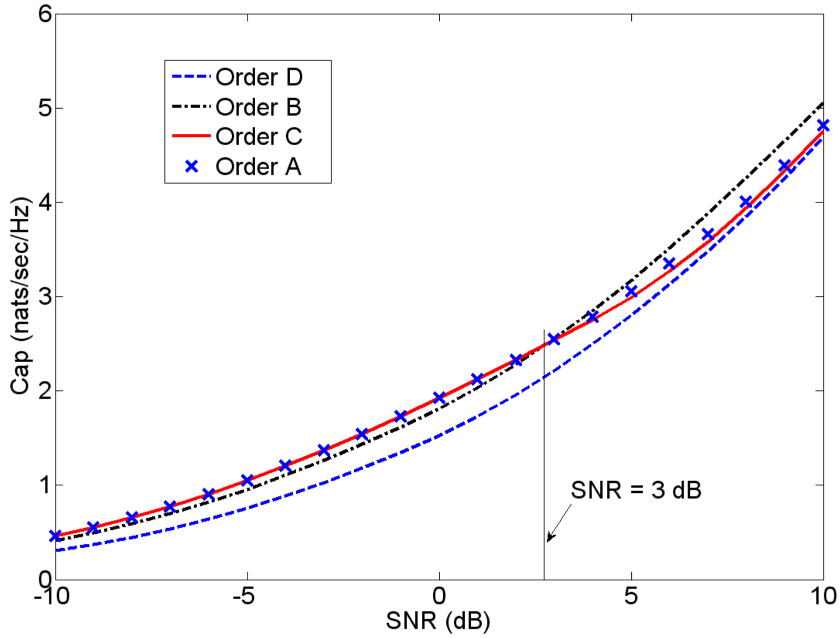

Figure 28: SNR of ordering (3x3).

7.3. Summary

The SNR gain of ordering was defined and studied. Closed-form expressions of the SNR gain of ordering under the IPRA were derived for the case of two Tx antennas. In the low SNR regime, the gain is SNR independent. In the intermediate SNR regime, it is SNR dependent and in the high SNR regime, it is a decreasing function of SNR. High gain of ordering can be achieved in the low and intermediate SNR regimes, while the gain is not that large in the high SNR regime, approaching one as $\gamma_0 \to \infty$.

8. Link to Multiple Access Wireless Channels

With the ever increasing development of cellular systems, multiuser communications is a topic that has gained a lot of importance in terms of research during the last years. In previous chapters, we studied the optimal ordering process for the coded V-BLAST in the context of point-to-point channels. In this section, we shift the focus to multiple access channels (MAC), i.e. the uplink of a cellular network where the base station uses multiple receive antennas to discriminate among the users which are in the same frequency (space-division multiple access (SDMA)), see Figure 29. More details about MAC can be found in [33].

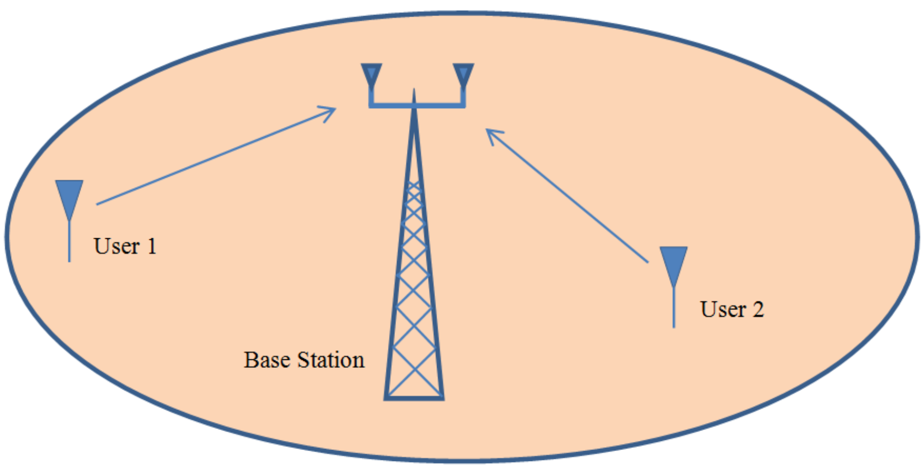

Figure 29: Uplink with single Tx antenna at each user and multiple Rx antennas at the base station.

Let us discuss the case of a cellular wireless system where the users have only one transmit antenna (e.g. due to complexity constraints). As the users are spread out over the cell, the signals sent out from the transmit antennas cannot be coordinated, so each user transmits independent signals. At the base station the signal of the users can be unmixed through the use of multiple receive antennas and appropriate signal processing. For example when detecting the signal corresponding to user i at the base station, we can first cancel the interference from the already detected users, project out the interference from the yet to be detected users and then apply MRC to the remaining signal plus correlated noise (correlated after the multiplication by the projection matrix). By doing this with each user, the multiple access channel is transformed into a set of virtually independent subchannels. This is in fact the V-BLAST architecture where each stream is interpreted as a user.

The system model can be expressed as in (3.2) i.e.

$$\mathbf{r} = \sum_{i=1}^{m} \mathbf{h}_i q_i + \boldsymbol{\xi} \tag{8.1}$$

The only difference is that now m is the number of users in the system (analogous to the number of Tx antennas in (3.2)) and $\mathbf{h}_i$ is a column vector containing the channel gains from the i-th user's Tx antenna to all Rx antennas of the base station, i.e. $\mathbf{h}_i$ is the SIMO channel corresponding to user i.

Since the users are geographically separated, their transmit signals arrive in different directions to the base station antenna array even when there is not a rich scattering environment, so the assumption of a full ranked channel matrix $\mathbf{H}$ is usually valid [33]. This is in contrast to the point-to-point case where a rich scattering environment is needed since all the transmit antennas are very close to each other. It follows then that all previous discussions about the stream detection order in the point-to-point coded V-BLAST can be applied to the multiple access wireless channels, but now instead of stream detection order we are dealing with user detection order.

Let us assume that we are in presence of a cellular network where the modulation and coding formats at the users ends can be modified in order to achieve a rate equal to the capacity of each user' channel (i.e. the IRA). In the hypothetical case where the base station is detecting the signal from only two users within the cell, we know from Theorem 1 that the optimum strategy is to detect last the user with the highest unprojected channel gain; this is in general the user situated closer to the base station. Note that from Theorem 4 the same holds true if optimum instantaneous power/rate allocation is used at each user transmission.

This strategy is optimal because it maximizes the sum rate of both users, so the total throughput of the system is maximized. However, from the point of view of the individual users, it can be seen as unfair because the user with the weakest unprojected channel gain has to deal with his bad channel condition and with the interference caused by the stronger user. Therefore, the rate at which this user can communicate will be severely affected. Another intrinsic drawback of the adaptive modulation and coding schemes is an increase in the system complexity and cost, since each user terminal must be designed to handle different coding and modulation strategies in an adaptive way. Modern communication systems already implement adaptive schemes, e.g. Wi-Max, Wi-Fi and LTE standards.

If we are looking for fairness and lower system complexity, a cellular network where the same coding and modulation formats are used for all users can be considered. The URA and the IPA are suitable for such a scenario. Under the URA all users transmit using the same rate and power. Under the IPA, although the transmission rate is uniform among the users, the power allocated to each user is different: users with weak channel gains enjoy higher transmission powers and, therefore, a better system performance is achieved[17]. Again in the hypothetical case where the base station is detecting the signal from only two users within the cell, either under URA or IPA, from Theorems 2 and 3, the optimum strategy is to detect first the user with the highest unprojected channel gain i.e. the user situated closer to the base station. In this case priority is given to the weakest user who is detected at the last step and, therefore, does not have to deal with the interference caused by the strongest user. This strategy, although better in terms of fairness and complexity, achieves a lower total system capacity.

[17] as compared to the system under the URA.

9. Conclusion

9.1. Summary of the thesis

Based on the fact that in the coded V-BLAST the instantaneous optimizations of the capacity and outage probability achieve the same lowest value of the outage probability, the problem of finding an optimal detection ordering was approached from the system capacity point of view. It was demonstrated that the optimum detection orderings under the IRA and the IPRA are SNR dependent while under the URA and the IPA are SNR independent. It was shown that the Foschini detection ordering is also optimum for the coded V-BLAST under the URA. Closed-form approximations were derived for the optimal detection ordering at low and high SNR under the IRA: based on approximations of the per-stream capacity equation, it was demonstrated that the optimum detection ordering maximizes the sum of the after detection channel gains at low SNR and that all detection ordering strategies offer approximately the same performance at high SNR. Therefore, at high SNR, is not worth the complexity effort of the ordering.

The optimum ordering strategies in the $(2 \text{ x } n)$ coded V-BLAST under the IRA, URA, IPA and the IPRA were established based on the before-processing channel gains. The analytical proofs of the optimum ordering strategies for these systems were given. Under the IRA and the IPRA, the best performance is achieved when the stream with highest before-processing channel gain is detected last. Furthermore, detecting this stream first is the best ordering strategy under the URA and the IPA.

Under the IRA, based on the $(2 \text{ x } n)$ results, necessary optimality conditions were derived for the general case. It was shown, based on Monte-Carlo simulations, that only a small percentage of the total possible orderings satisfy the conditions and that this percentage decreases very fast with the number of Tx antennas. In the best case scenario, only one ordering satisfies them (this is a very likely scenario). Although the problem of finding the maximum number of orderings that can satisfy the necessary optimality conditions remains open, based on the definition of independent necessary conditions, this maximum number was lower and upper bounded.

The SNR gain of ordering was defined. A closed-form analysis of the SNR gain of ordering for the case of two Tx antennas under the IPRA was provided. At low and intermediate SNR the gain offered by the optimal ordering is significant, but at high SNR the ordering does not offer a significant improvement to the system performance, and thus is not worth the complexity effort.

Suboptimal detection orderings were proposed and their performances were shown to be very close to the optimum one. It was shown that the results found from the point-to-point perspective are also applicable to the multiuser communications, i.e. multiple access channels.

9.2. Future research

One limitation of this research is that the closed-form analysis presented is limited to the ZF V-BLAST. One possible extension would be to generalize these results to the MMSE V-BLAST, which is the capacity achieving strategy.

Future research is needed in some areas. First, no loss of optimality by considering the WF to obtain the optimum detection ordering under the IPRA was observed numerically, but it lacks an analytical proof. Second, the exact maximum number of combinations that can satisfy the necessary optimality conditions under the IRA is still an open (hard combinatorial) problem. Third, the analytical results of the SNR gain are limited to the case of two Tx antennas, this closed form analysis can be extended to $m > 2$.

Finally, it should be pointed out that the above results have been derived under ideal conditions. An important topic for future research is to investigate the impacts of practical conditions, such as imperfect channel estimation. Hence, this work is only a partial analytical framework to the full understanding of the optimal ordering in the coded V-BLAST.

10. References

MIMO channel capacity and BLAST architecture

[1] I. E. Telatar, “Capacity of Multi-Antenna Gaussian Channels,” *AT&T Bell Lab. Internal Tech. Memo.*, Jun. 1995 (*European Trans. Telecom.*, v. 10, N. 6, Dec. 1999).

[2] G. J. Foschini, “Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multiple Antennas,” *Bell Lab. Tech. J.*, v. 1, N. 2, pp. 41-59, 1996.

[3] G. J. Foschini, P. W. Wolnainsky, G. D. Golden, and R. A. Valenzuela, “Simplified Processing for High Spectral Efficiency Wireless Communication Employing Multi-Element Arrays,” *IEEE J. Select. Areas Commun.*, vol. 17, no. 11, pp. 1841–1852, Nov. 1999.

[4] P. W. Wolnainsky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, “V-BLAST: An Architecture for Achieving Very High Data Rates Over the Rich-Scattering Wireless Channel,” *Int. Symposium on Signals, Systems and Electronics* (*ISSSE)*, Pisa, Italy, 1998.

Reducing V-BLAST complexity

[5] W. K. Wai, C.-Y. Tsui, R. S.Cheng, “A Low Complexity Architecture for the V-BLAST System,” *IEEE Wireless Commun. and Networking Conf. (WCNC)*, Chicago, IL, USA, v. 1, pp. 310-314, Sep. 2000.

[6] J. Benesty, Y. Huang, and J. Chen, “A fast Recursive Algorithm for Optimum Sequential Signal Detection in a BLAST System,” *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp. 1722-1730, July 2003.

[7] Z. Luo, S. Liu, M. Zhao, and Y. Liu, “ A Novel Fast Recursive MMSE-SIC Detection Algorithm for V-BLAST Systems”, *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, pp. 2022-2025, June 2007.

[8] J. Chen, S. Jin, and Y. Wang, “Reduced Complexity MMSE-SIC Detector in V-BLAST Systems,” *IEEE Personal, Indoor and Mobile Radio Commun. (PIMRC)*,pp. 1–5, Sep. 2007.

[9] B. Hassibi, “An Efficient Square-Root Algorithm for BLAST,” in *Proc. IEEE Int. Conf. Acoustics Speech, and Signal Processing, (ICASSP)*, pp. 737–740, June 2000.

[10] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, “Efficient Algorithm for Decoding Layered Space-Time Codes,” *IEEE Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, Oct. 2001.

[11] R. Bohnke, D. Wubben, V. Kuhn, and K. D. Kammeyer, “Reduced Complexity MMSE Detection for BLAST Architectures,” *IEEE Global Telecom. Conf. (GLOBECOM)*, vol. 4. pp. 2258–2262, Dec. 2003.

[12] H. Zhu, Z. Lei, and F. P. S. Chin, “An Improved Square-Root Algorithm for BLAST,” *IEEE Signal Process Lett.*, vol. 11, no. 9, pp. 772-775, Sep. 2004.

[13] H. Zhu, W. Chen, B. Li, and F. Gao,” An Improved Square-Root Algorithm for V-BLAST Based on Efficient Inverse Cholesky Factorization,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 1, pp. 43-48, Jan. 2011.

Performance analysis of the V-BLAST

[14] S. Loyka and F. Gagnon, “Performance Analysis of the V-BLAST Algorithm: An Analytical Approach,” *IEEE Trans.Wireless Commun.*, vol.3, no. 4, pp. 1326–1337, Jul. 2004.

[15] Y. Jiang, X. Zheng, and J. Li, “Asymptotic Performance Analysis of V-BLAST,” *IEEE Global Telecom. Conf. (GLOBECOM)*, pp. 3882-3886, 2005.

[16] S. Loyka and F. Gagnon, “V-BLAST Without Optimal Ordering: Analytical Performance Evaluation for Rayleigh Fading Channels,” *IEEE Trans. Commun.*, vol. 54, no. 6, pp. 1109-1120, June 2006.

[17] H. Zhang, H. Dai, and B. L. Hughes, “On the Diversity-Multiplexing Tradeoff for Ordered SIC Receivers Over MIMO Channels,” *IEEE Int. Conf. on Commun. (ICC)*, pp. 1368-1373, 2006.

[18] R. Xu and F. C. M. Lau, “A Novel Approach to Analyzing V-BLAST MIMO Systems With Two Transmit Antennas”, *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1591-1595, May 2007.

[19] S. Loyka and F. Gagnon, “On Outage and Error Rate Analysis of the Ordered V-BLAST,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3679-3685, Oct. 2008.

[20] S. H. Nam *et al.*, “Transmit Power Allocation for a Modified V-BLAST System,” *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1074–1079, July 2004.

[21] N. Wang and S. D. Blostein, “Minimum BER Transmit Power Allocation for MIMO Spatial Multiplexing Systems,” *IEEE International Conference on Commun.*, San Diego, vol. 4, pp. 2282– 2286, May 2005.

[22] 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. Lett.*, vol. 9, no. 9, pp. 826–828, Sept. 2005.

[23] N. Wang and S. D. Blostein, “Minimum BER Transmit Power Allocation and Beamforming for Two-Input Multiple-Output Spatial Multiplexing Systems,” *IEEE Trans. Vehicular Technology*, vol. 56, no. 2, pp. 1591-1595, March 2007.

[24] V. Kostina and S. Loyka, “On Optimum Power Allocation for the V-BLAST,” *IEEE Trans. Commun.*, v. 56, N. 6, pp. 999-1012, June 2008.

[25] Y Jiang, M. K. Varanasi and J. Li, “Performance Analysis of ZF and MMSE Equalizers for MIMO Systems: An In-Depth Study of the High SNR Regime,” *IEEE Trans. Information Theory*, v. 57, no. 4, pp. 2008-2026, Apr. 2011.

V-BLAST under channel estimation errors

[26] X. Zhang and B. Ottersten, “Performance Analysis of V-BLAST Structure With Channel Estimation Errors,” *IEEE Signal Processing Advances in Wireless Commun. (SPAWC)* , pp. 487–491, June 2003.

[27] R. Narasimhan, “Error Propagation Analysis of V-BLAST With Channel-Estimation Errors”, *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 27-31, Jan. 2005.

[28] Ch. Wang *et al*., “On the Performance of the MIMO Zero-Forcing Receiver in the Presence of Channel Estimation Error”, *IEEE Trans. Wireless Commun.*, vol. 6, no. 3, pp. 805-810, March 2007.

[29] J. Chen and X. Yu, “ZF V-BLAST for Imperfect MIMO Channels Using Average Performance Optimization,” . *IEEE Int. Conf. Acoustics Speech, and Signal Processing (ICASSP)*, pp. III–141-III-144, Apr. 2007.

[30] W. Peng *et al.*, “Effects of Channel Estimation Errors on V-BLAST Detection,” *IEEE Global Telecom. Conf. (GLOBECOM)*, pp. 1-5, Dec. 2008.

Coded V-BLAST

[31] V. Kostina and S. Loyka, “Optimum Power and Rate Allocation for Coded V-BLAST: Average Optimization," *IEEE Trans. Commun.*, v. 59, no. 3, pp. 877-887, Mar. 2011.

[32] V. Kostina and S. Loyka, “Optimum Power and Rate Allocation for Coded V-BLAST: Instantaneous Optimization,” *IEEE Trans. Commun.*, v. 59, no. 10, pp. 2841-2850, Oct. 2011.

Books

[33] D. Tse and P. Viswanath, “Fundamentals of Wireless Communications,” Cambridge University Press, 2005.

[34] F. R. Gantmacher, “The Theory of Matrices,” Volume 1, Chelsea Publishing Company, 1959.

11. Appendixes

Appendix A: Waterfilling algorithm

A parallel additive white Gaussian noise (AWGN) channel is a channel consisting in a set of non-interfering sub-channels each of which is corrupted by AWGN. A natural strategy for reliable communication over a parallel AWGN channel is to allocate power to each sub-channel and use separate capacity achieving temporal codes to communicate over each of the sub-channels. The maximum rate of reliable communication under this scheme is

$$\sum_{i=1}^{m} \ln\left(1+\alpha_i g_i \gamma_0\right) \tag{11.1}$$

where m is the number of sub-channels, α_i and g_i are the power and the channel gain corresponding sub-channel i and γ_0 is the average SNR at the output of each sub-channel [33]. Furthermore the power allocation can be chosen to maximize the rate in (11.1).

The optimum power allocation strategy to transmit through a parallel AWGN channel when channel state information is available at the transmitter is the waterfilling (WF) algorithm [33]. The power allocation under the WF algorithm is given by

$$\alpha_i = \left[\mu - \frac{1}{\gamma_0 g_i}\right]_+ \tag{11.2}$$

where $[x]_+ = \max[x,0]$ and μ is the water level. μ is chosen such that the total power constraint is met i.e. $\sum_{i=1}^{m} \alpha_i = m$. The principle of this power allocation strategy is to allocate more power to the stronger streams taking advantage of the better channel conditions and less or even no power to the weaker ones. Figure 30 gives a pictorial view of this strategy. It is not difficult to show that the total system capacity under the WF algorithm (given that all streams are active) can be expressed as a function of the water level as

$$C_{WF} = \sum_{i=1}^{m} \ln\left(\mu g_i \gamma_0\right) \tag{11.3}$$

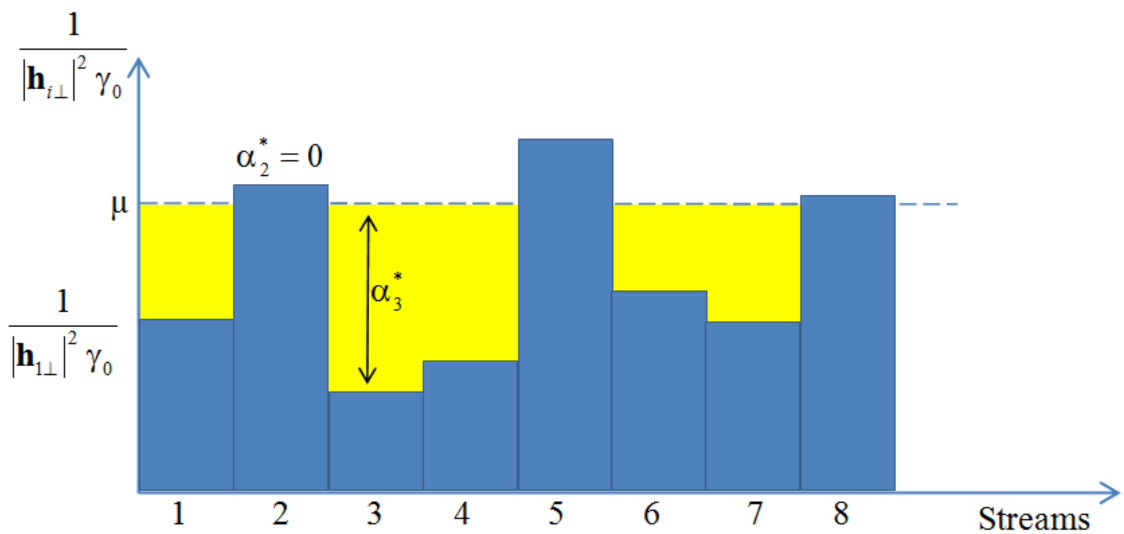

Figure 30: Pictorial representation of the waterfilling algorithm.

Two active streams

When two streams are active $(m^* = 2)$ [18], the system capacity is

$$C = \ln(1+\alpha_1 g_1 \gamma_0) + \ln(1+\alpha_2 g_2 \gamma_0) \tag{11.4}$$

The power allocated to the streams are given by

$$\alpha_i = \mu - \frac{1}{\gamma_0 g_i}, \quad i = 1, 2 \tag{11.5}$$

and μ can be found from the total power constraint as follows :

$$\alpha_1 + \alpha_2 = 2 \tag{11.6}$$

$$\Rightarrow \mu = 1 + \frac{1}{2\gamma_0}\left(\frac{1}{g_1} + \frac{1}{g_2}\right) \tag{11.7}$$

where equation (11.6) states the total power constraint for the case of two active streams. Equation (11.7) follows from substituting the WF power allocation (11.5) into (11.6) and

[18] m^* is the number of active streams.

some straightforward mathematical manipulations. Substituting (11.7) into (11.5) the powers α_1 and α_2 can be expressed as a function of the channel gains as

$$\alpha_1 = 1 - \frac{1}{2g_1\gamma_0} + \frac{1}{2g_2\gamma_0}, \quad \alpha_2 = 1 - \frac{1}{2g_2\gamma_0} + \frac{1}{2g_1\gamma_0} \tag{11.8}$$

Note that substituting (11.8) into (11.4), after some straightforward manipulations the system capacity is expressed as

$$C = \ln\left(\mu g_1\gamma_0\right) + \ln\left(\mu g_2\gamma_0\right) \tag{11.9}$$

On the other hand if the following condition is satisfied

$$\mu = 1 + \frac{1}{2\gamma_0}\left(\frac{1}{g_1} + \frac{1}{g_2}\right) \leq \max\left[\frac{1}{\gamma_0 g_1}, \frac{1}{\gamma_0 g_2}\right] \tag{11.10}$$

the weakest stream is tuned off and all the available power is assigned to the strongest one ($m^* = 1$). Equation (11.10) follows from the WF power allocation formula (11.2). For the case of $m^* = 1$ the system capacity is expressed as

$$C = \ln\left(1 + 2\gamma_0 \max\left[g_1, g_2\right]\right) \tag{11.11}$$

Appendix B: Optimum ordering under the IPRA (m=2)

Note: The derivations in this appendix make use of results derived in Appendix A (WF algorithm).

Proof of Theorem 4: For the case of two Tx antennas, the channel matrix is given by $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2]$. The two possible orderings are $\pi_1 = \{1,2\}$ and $\pi_2 = \{2,1\}$.

From the WF algorithm, when the channel gain of a given stream is too weak for it to be valuable to transmit information, that stream is turned off by assigning zero power to it, so SNR regimes can be defined for the case of two Tx antennas:

- ***Low SNR regime:*** Both orderings have one active stream
- ***Intermediate SNR regime:*** The optimum ordering has one active stream and the suboptimum ordering has two active streams.
- ***High SNR regime:*** Both orderings have two active streams.

The proof of theorem 4 will be based on proving that it holds for these tree SNR regimes.

Low SNR regime

Statement 1: In the low SNR regime, the optimum ordering strategy is to detect the stream with highest before-detection channel gain at the last step:

$$\pi^* = \arg\max_{\pi} C_{IPRA}(\pi) = \{1,2\} \quad \text{iff} \quad |\mathbf{h}_1| < |\mathbf{h}_2| \qquad (11.12)$$

The "only if" part in (11.12) is true when $\varphi \neq \pi/2$. When $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ any ordering is optimum.

Proof: In the case of two transmit antennas, as a result of the WF algorithm, when one antenna is turned off, all the available power is assigned to the strongest one. The total power constraint is $\sum_{i=1}^{2}\alpha_i(\pi)=2$, and the capacities of these orderings in the low SNR regime are given by:

$$C_{WF}(\pi_1)=\ln\left(1+2\max\left[\left|\mathbf{h}_{1\perp 2}\right|^2,\left|\mathbf{h}_2\right|^2\right]\gamma_0\right) \tag{11.13}$$

$$C_{WF}(\pi_2)=\ln\left(1+2\max\left[\left|\mathbf{h}_{2\perp 1}\right|^2,\left|\mathbf{h}_1\right|^2\right]\gamma_0\right) \tag{11.14}$$

where $\mathbf{h}_{i\perp j}$ refers to the projection of $\mathbf{h}_i$ orthogonal to $\mathbf{h}_j$ and γ_0 is the average SNR at each receive antenna. Equations (11.13) and (11.14) follow from making the appropriate substitutions of the channel gains into (11.11) depending on the specific ordering. To prove the "if" part, assume that $C_{WF}(\pi_1)>C_{WF}(\pi_2)$ so that

$$\max\left[\left|\mathbf{h}_{1\perp 2}\right|^2,\left|\mathbf{h}_2\right|^2\right]>\max\left[\left|\mathbf{h}_{2\perp 1}\right|^2,\left|\mathbf{h}_1\right|^2\right] \tag{11.15}$$

$$\Rightarrow \max\left[\left|\mathbf{h}_{1\perp 2}\right|^2,\left|\mathbf{h}_2\right|^2\right]=\left|\mathbf{h}_2\right|^2 \tag{11.16}$$

$$\Rightarrow \left|\mathbf{h}_2\right|^2>\left|\mathbf{h}_1\right|^2 \tag{11.17}$$

The inequality (11.15) follows from (11.13), (11.14) and the fact that $\ln x$ is an increasing function of x. The equality (11.16) follows from (11.15) and from noting that the projection of a vector cannot increase its norm (please note that if $\max\left[\left|\mathbf{h}_{1\perp 2}\right|^2,\left|\mathbf{h}_2\right|^2\right]=\left|\mathbf{h}_{1\perp 2}\right|^2$ inequality (11.15) is contradicted since $\left|\mathbf{h}_{1\perp 2}\right|^2\le\left|\mathbf{h}_1\right|^2$). Inequality (11.17) follows from substituting (11.16) into (11.15). Now, to prove the "only if" part, assume that (11.17) holds so that

$$\max\left[\left|\mathbf{h}_{1\perp 2}\right|^2,\left|\mathbf{h}_2\right|^2\right]=\left|\mathbf{h}_2\right|^2>\max\left[\left|\mathbf{h}_{2\perp 1}\right|^2,\left|\mathbf{h}_1\right|^2\right] \tag{11.18}$$

$$\Rightarrow C_{WF}(\pi_1)>C_{WF}(\pi_2) \tag{11.19}$$

The equality and the inequality in (11.18) follow from $\left|\mathbf{h}_2\right|>\left|\mathbf{h}_1\right|$ and the fact that a projection cannot increase the norm of a vector. Inequality (11.19) follows immediately from (11.18), (11.13) and (11.14). Equation (11.12) follows. Note that if $\varphi=\pi/2$ and/or

$|\mathbf{h}_1| = |\mathbf{h}_2|$, the inequality (11.15) becomes an equality showing that any ordering is optimum under these circumstances. ■

High SNR regime

Statement 2: In the high SNR regime, the optimum ordering strategy is to detect the stream with highest before-detection channel gain at the last step:

$$\pi^* = \arg\max_{\pi} C_{IPRA}(\pi) = \{1,2\} \quad \text{iff} \quad |\mathbf{h}_1| < |\mathbf{h}_2| \tag{11.20}$$

The "only if" part in (11.20) is true when $\varphi \neq \pi/2$. When $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ any ordering is optimum.

Proof: In the high SNR regime both orderings have two active streams. The capacities are given by

$$C_{WF}(\pi_1) = \ln\left(1+\alpha_1^* |\mathbf{h}_{1\perp 2}|^2 \gamma_0\right) + \ln\left(1+\alpha_2^* |\mathbf{h}_2|^2 \gamma_0\right) \tag{11.21}$$

$$C_{WF}(\pi_2) = \ln\left(1+\alpha_a^* |\mathbf{h}_{2\perp 1}|^2 \gamma_0\right) + \ln\left(1+\alpha_b^* |\mathbf{h}_1|^2 \gamma_0\right) \tag{11.22}$$

where the pairs α_1^*, α_2^* and α_a^*, α_b^* are the optimum WF power allocations for the orderings π_1 and π_2 respectively. Equations (11.21) and (11.22) can be expressed as a function of their respective water levels as follows:

$$C_{WF}(\pi_1) = \ln\left(\mu_1^2 |\mathbf{h}_{1\perp 2}|^2 |\mathbf{h}_2|^2 \gamma_0^2\right) \tag{11.23}$$

$$C_{WF}(\pi_2) = \ln\left(\mu_2^2 |\mathbf{h}_{2\perp 1}|^2 |\mathbf{h}_1|^2 \gamma_0^2\right) \tag{11.24}$$

Equations (11.23) and (11.24) follow from making the appropriate substitutions of the channel gains and the water level in (11.9) depending on the specific ordering. The water levels μ_1 and μ_2 are expressed as

$$\mu_1 = 1 + \frac{1}{2\gamma_0}\left(\frac{1}{|\mathbf{h}_{1\perp 2}|^2} + \frac{1}{|\mathbf{h}_2|^2}\right) \tag{11.25}$$

$$\mu_2 = 1 + \frac{1}{2\gamma_0}\left(\frac{1}{\left|\mathbf{h}_{2\perp 1}\right|^2} + \frac{1}{\left|\mathbf{h}_1\right|^2}\right) \tag{11.26}$$

Equations (11.25) and (11.26) follow from (11.7) after the appropriate substitution of the channels gains depending on the ordering. To prove the "if" part assume that

$$C_{WF}\left(\pi_1\right) > C_{WF}\left(\pi_2\right) \tag{11.27}$$

so that the following set of inequalities holds

$$\mu_1^2 \left|\mathbf{h}_1\right|^2 \left|\mathbf{h}_2\right|^2 \gamma_0^2 \sin^2\varphi > \mu_2^2 \left|\mathbf{h}_1\right|^2 \left|\mathbf{h}_2\right|^2 \gamma_0^2 \sin^2\varphi \tag{11.28}$$

$$\Rightarrow \mu_1 > \mu_2 \tag{11.29}$$

$$\Rightarrow 1 + \frac{1}{2\gamma_0}\left(\frac{1}{\left|\mathbf{h}_1\right|^2 \sin^2\varphi} + \frac{1}{\left|\mathbf{h}_2\right|^2}\right) > 1 + \frac{1}{2\gamma_0}\left(\frac{1}{\left|\mathbf{h}_2\right|^2 \sin^2\varphi} + \frac{1}{\left|\mathbf{h}_1\right|^2}\right) \tag{11.30}$$

$$\Rightarrow \left|\mathbf{h}_1\right|^2 \sin^2\varphi + \left|\mathbf{h}_2\right|^2 > \left|\mathbf{h}_2\right|^2 \sin^2\varphi + \left|\mathbf{h}_1\right|^2 \tag{11.31}$$

$$\Rightarrow \left|\mathbf{h}_2\right|^2 \left(1 - \sin^2\varphi\right) > \left|\mathbf{h}_1\right|^2 \left(1 - \sin^2\varphi\right) \tag{11.32}$$

$$\Rightarrow \left|\mathbf{h}_2\right|^2 > \left|\mathbf{h}_1\right|^2 \tag{11.33}$$

The inequality in (11.28) follows from three steps: the substitution of (11.23) and (11.24) into (11.27), the use of the geometrical representation (4.14) and the fact that $\ln(x)$ is an increasing function of x. Inequality (11.29) follows immediately from (11.28). Inequality (11.30) results from substituting (11.25) and (11.26) into (11.29) and the use of the geometrical representation (4.14). Inequalities (11.31), (11.32) and (11.33) are the result of straightforward mathematical manipulations. The "only if" part is proved after noting that the same chain of inequalities holds in the reverse direction i.e. starting from (11.33) and ending in (11.27). Equation (11.20) follows. Note that if $\varphi = \pi/2$ and/or $\left|\mathbf{h}_1\right| = \left|\mathbf{h}_2\right|$, the inequality (11.30) becomes an equality showing that any ordering is optimum under these circumstances. ■

Intermediate SNR regime

Statement 3: In the intermediate SNR regime, the optimum ordering strategy is to detect the stream with highest before detection channel gain at the last step:

$$\pi^* = \arg\max C_{IPRA}(\pi) = \{1,2\} \quad \text{iff} \quad |\mathbf{h}_1| < |\mathbf{h}_2| \tag{11.34}$$

Proof: Recall from section 6.2.3 that the SNR threshold at which the suboptimum ordering starts using two active streams is lower than the threshold corresponding to the optimum ordering[19]. Therefore, in this regime the optimum ordering has one active stream and the suboptimum ordering has two active streams. The capacity equations are given by (11.13) and (11.22) respectively, i.e.

$$C_{WF}(\pi_1) = \ln\left(1 + 2\max\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right]\gamma_0\right) \tag{11.35}$$

$$C_{WF}(\pi_2) = \ln\left(1 + \alpha_a^* |\mathbf{h}_{2\perp 1}|^2 \gamma_0\right) + \ln\left(1 + \alpha_b^* |\mathbf{h}_1|^2 \gamma_0\right) \tag{11.36}$$

Equation (11.35) can be reduced to

$$C_{WF}(\pi_1) = \ln\left(1 + 2|\mathbf{h}_2|^2 \gamma_0\right) \tag{11.37}$$

as it is shown below. Ordering π_1 has a better performance than π_2 provided that

$$C_{WF}(\pi_1) > C_{WF}(\pi_2) \tag{11.38}$$

Equation (11.37) will be proved showing that if $\max\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right] = |\mathbf{h}_{1\perp 2}|^2$ then equation (11.38) is false. The following set of equations illustrates this. Assuming that

$$\max\left[|\mathbf{h}_{1\perp 2}|^2, |\mathbf{h}_2|^2\right] = |\mathbf{h}_{1\perp 2}|^2 \tag{11.39}$$

then

[19] Recall also that if $\varphi = \pi/2$ and/or $|\mathbf{h}_1| = |\mathbf{h}_2|$ this SNR regime does not exist. Therefore in the analysis below $|\mathbf{h}_1| \neq |\mathbf{h}_2|$ and $\varphi \neq \pi/2$ is assumed.

$$\left|\mathbf{h}_{2\perp 1}\right|^2 < \left|\mathbf{h}_2\right|^2 < \left|\mathbf{h}_{1\perp 2}\right|^2 < \left|\mathbf{h}_1\right|^2 \tag{11.40}$$

$$\Rightarrow \max\left[\left|\mathbf{h}_{2\perp 1}\right|^2, \left|\mathbf{h}_1\right|^2\right] = \left|\mathbf{h}_1\right|^2 \tag{11.41}$$

$$\Rightarrow C_{WF}\left(\pi_1\right) < C_{WF}\left(\pi_2\right)_1 \le C_{WF}\left(\pi_2\right) \tag{11.42}$$

where $C_{WF}\left(\pi_2\right)_1$ is the capacity offered by π_2 with only one active stream given by (11.14). The set of inequalities in (11.40) follows from (11.39) and the fact that the projection of a vector cannot increase its norm[20]. Equation (11.41) follows from the set of inequalities in (11.40). The first inequality in (11.42) follows from substituting (11.39) and (11.41) into (11.35) and (11.14) respectively and the fact that $\ln x$ is an increasing function of x. The second inequality in (11.42) is because in the intermediate SNR regime the optimum WF power allocation strategy for π_2 is to allocate power to both streams; hence a strategy where only one stream is active is suboptimal. The inequalities in (11.42) contradicts (11.38), so that

$$\max\left[\left|\mathbf{h}_{1\perp 2}\right|^2, \left|\mathbf{h}_2\right|^2\right] = \left|\mathbf{h}_2\right|^2 \tag{11.43}$$

and (11.37) follows.

Now let us define $C_{WF}\left(\pi_1\right)'$ as the capacity offered by π_1 using the optimum WF power allocation for π_2 when both streams are active i.e.

$$C_{WF}\left(\pi_1\right)' = \ln\left(1+\alpha_a^*\left|\mathbf{h}_{1\perp 2}\right|^2\gamma_0\right) + \ln\left(1+\alpha_b^*\left|\mathbf{h}_2\right|^2\gamma_0\right) \tag{11.44}$$

The following inequality is evident,

$$C_{WF}\left(\pi_1\right)' \le C_{WF}\left(\pi_1\right) \tag{11.45}$$

since $C_{WF}\left(\pi_1\right)$, from the WF algorithm, is the maximum capacity offered by π_1. Now it will be shown that

[20] Furthermore, the case of $\varphi = \pi/2$ is not considered since, in this case, the intermediate SNR regime does not exist.

$$C_{WF}(\pi_1)' > C_{WF}(\pi_2) \Leftrightarrow |\mathbf{h}_1| < |\mathbf{h}_2| \tag{11.46}$$

and based on (11.45) it will be concluded that Statement 3 is proved. Equation (11.44) can be expressed as

$$C_{WF}(\pi_1)' = \ln\left[\left(1+\left(1-\frac{1}{2g_2\beta}+\frac{1}{2g_1}\right)g_1\beta\right)\left(1+\left(1-\frac{1}{2g_1}+\frac{1}{2g_2\beta}\right)g_2\right)\right] \tag{11.47}$$

where $g_i = |\mathbf{h}_i|^2 \gamma_0$ and $\beta = \sin^2\varphi$ and the allocated powers α_a^* and α_b^* have been expressed as a function of the channel gains i.e.

$$\alpha_a^* = 1 - \frac{1}{2|\mathbf{h}_2|^2 \sin^2\varphi\gamma_0} + \frac{1}{2|\mathbf{h}_1|^2 \gamma_0} \tag{11.48}$$

$$\alpha_b^* = 1 - \frac{1}{2|\mathbf{h}_1|^2 \gamma_0} + \frac{1}{2|\mathbf{h}_2|^2 \sin^2\varphi\gamma_0} \tag{11.49}$$

Equations (11.48) and (11.49) follow from (11.8) after the appropriate substitution of the channel gains and the use of the geometrical representation (4.14). After some manipulations, (11.47) can be expressed as

$$C_{WF}(\pi_1)' = \ln\left[\left(\frac{2g_2+A}{2g_2}\right)\left(\frac{2g_1\beta+B}{2g_1\beta}\right)\right] \tag{11.50}$$

where A and B are given by

$$A = 2g_1g_2\beta - g_1 + g_2\beta \tag{11.51}$$

$$B = 2g_1g_2\beta - g_2\beta + g_1 \tag{11.52}$$

Using $g_i = |\mathbf{h}_i|^2 \gamma_0$, $\beta = \sin^2\varphi$, (11.48) and (11.49), equation (11.36) can be expressed as

$$C_{WF}(\pi_2) = \ln\left[\left(1+\left(1-\frac{1}{2g_2\beta}+\frac{1}{2g_1}\right)g_2\beta\right)\left(1+\left(1-\frac{1}{2g_1}+\frac{1}{2g_2\beta}\right)g_1\right)\right] \tag{11.53}$$

$$= \ln\left[\left(\frac{2g_1+A}{2g_1}\right)\left(\frac{2g_2\beta+B}{2g_2\beta}\right)\right] \tag{11.54}$$

where A and B are given by (11.51) and (11.52) respectively. If ordering π_1, under the WF power allocation corresponding to π_2, has a better performance than π_2 under optimum power allocation,

$$C_{WF}(\pi_1)' > C_{WF}(\pi_2) \tag{11.55}$$

then, the following set of inequalities holds

$$\left(\frac{2g_2 + A}{2g_2}\right)\left(\frac{2g_1\beta + B}{2g_1\beta}\right) > \left(\frac{2g_1 + A}{2g_1}\right)\left(\frac{2g_2\beta + B}{2g_2\beta}\right) \tag{11.56}$$

$$\Rightarrow (2g_2 + A)(2g_1\beta + B) > (2g_1 + A)(2g_2\beta + B) \tag{11.57}$$

$$\Rightarrow g_2(B - A\beta) > g_1(B - A\beta) \tag{11.58}$$

$$\Rightarrow g_2 > g_1 \Leftrightarrow |\mathbf{h}_2| > |\mathbf{h}_1| \tag{11.59}$$

Inequality (11.56) follows from substituting (11.50) and (11.54) into (11.55) and noting that $\ln x$ is an increasing function of x. Inequalities (11.57), (11.58), (11.57) and (11.59) follow from straightforward mathematical manipulations. It is evident that the same chain of inequalities holds in the reverse direction i.e. starting in (11.59) and ending in (11.55), therefore, equation (11.46) follows. Finally, equation (11.34) follows from (11.45) and (11.46).

It can be concluded that from (11.12), (11.20) and (11.34) equation (6.30) follows. ■

Appendix C: Matlab codes

IRA: Principal function.

```
function principal_IRA
%
%principal.m   Returns the CDFs of the Inverse, the Max Sum Capacity of the Last Two steps
%              (MSCL2S), the Max Sum Powers of the Last Two Steps (MSPL2S), the Max Sum
%              Powers (MSP), the optimum detection orderings and the unordered detection.
%              It Also generates a descriptive graph for each ordering and compares their
%              performances.

% initializing data

SNR_dB=10;                  %SNR value (dB)
m=3;                        %number of transmit antennas.
n=3;                        %number of receive antennas.
M=1000;                     %number of channel realizations.
snr=10^(SNR_dB/10);         %received SNR.10^(dB/10).
OptCap_l=zeros(1,M);        %vector to store the capacity of the optimum ordering.
Inverse_CapB_l=zeros(1,M);%vector to store the capacity of the Inverse ordering.
MSPL2S_CapA_l=zeros(1,M); %vector to store the capacity of the MSPL2S ordering.
MSCL2S_CapC_l=zeros(1,M); %vector to store the capacity of the MSCL2S ordering.
MSP_CapD_l=zeros(1,M);      %vector to store the capacity of the MSP ordering.
ConvCap_l=zeros(1,M);       %vector to store the capacity of the unordered detection.
VAL=0;                      %counter to follow the validation of the code.
tic;
fprintf('\nPlease wait, Matlab is running the simulation ...\n')

% generating M channel realizations and obtaining the capacity offered by each ordering
strategy

for l=1:M
    clear OptCap Inverse_CapB MSPL2S_CapA  MSCL2S_CapC MSP_CapD H Validated Hopt ConvCap;
    H = 1/sqrt(2)*(randn(n,m) + 1i*randn(n,m)); % random channel matrix, with entries iid CN-
(0,1).
    [OptCap Validated Hopt MSP_CapD ConvCap] = bruta_IRA(n, m, H, snr);    % Returns the
maximum capacity, the capacity of the unordered detection, and the capacity of the MSP
ordering for a given channel realization.

    Inverse_CapB = cap_Inverse_ordering(n, m, H, snr);
    MSPL2S_CapA = cap_MSPL2S_ordering(n, m, H, snr);
    MSCL2S_CapC = cap_MSCL2S_ordering(n, m, H, snr);

    %Saving the capacity of each ordering for the given channel realization

    OptCap_l(l)=OptCap;
    Inverse_CapB_l(l)=Inverse_CapB;
    MSPL2S_CapA_l(l)=MSPL2S_CapA;
    MSCL2S_CapC_l(l)=MSCL2S_CapC;
    MSP_CapD_l(l)=MSP_CapD;
    ConvCap_l(l)=ConvCap;

    if Validated ==1, VAL = VAL+1;    %Counting the validation of the code for each cannel
realization.
    end
end

Cap_Med=sum(OptCap_l)/M; %Obtaining the mean capacity

%collecting the difference between the optimum capacity and the capacity offered by each
ordering strategy

DIFF_A = (OptCap_l-MSPL2S_CapA_l);
DIFF_B = (OptCap_l-Inverse_CapB_l);
DIFF_C = (OptCap_l-MSCL2S_CapC_l);
DIFF_D=(OptCap_l-MSP_CapD_l);
DIFF_UNORD=(OptCap_l-ConvCap_l);
```

```
%normalizing the difference by the mean capacity

DIFF_A_div_Cap_Med=DIFF_A/Cap_Med;
DIFF_B_div_Cap_Med=DIFF_B/Cap_Med;
DIFF_C_div_Cap_Med=DIFF_C/Cap_Med;
DIFF_D_div_Cap_Med=DIFF_D/Cap_Med;
DIFF_UNORD_div_Cap_Med=DIFF_UNORD/Cap_Med;

%expressing the difference between the opt cap and the capacity offered by each ordering
strategy in percent

PRCENT_CAP_LOSS_A=(DIFF_A./OptCap_l)*100;
PRCENT_CAP_LOSS_B=(DIFF_B./OptCap_l)*100;
PRCENT_CAP_LOSS_C=(DIFF_C./OptCap_l)*100;
PRCENT_CAP_LOSS_D=(DIFF_D./OptCap_l)*100;
PRCENT_CAP_LOSS_UNORD=(DIFF_UNORD./OptCap_l)*100;

%obtaining the channels that did not achieve the maximum capacity using each ordering
strategy

F_DIFF_A=find(DIFF_A);
F_DIFF_B=find(DIFF_B);
F_DIFF_C=find(DIFF_C);
F_DIFF_D=find(DIFF_D);
F_DIFF_UNORD=find(DIFF_UNORD);

%histogram and cumulative sum of the optimum ordering

n_cap_opt=[0:(max(OptCap_l)/50):max(OptCap_l)];
n_elements_cap_opt=hist(OptCap_l,n_cap_opt);    %obtaining the histogram of the optimum
ordering capacity.
c_elements_cap_opt=cumsum(n_elements_cap_opt);  %obtaining the cumulative sum of the optimum
ordering capacity.

%Figures describing the performance of each ordering strategy

figure('Name','Inverse Ordering','NumberTitle','on')
if isempty(F_DIFF_B),
else
[nb c_elements_b_mod c_elements_b n_cap_b c_elements_cap_b
AVG_PRCENT_CAP_LOSS_MOD_B]=plot_figures(F_DIFF_B, OptCap_l, DIFF_B,
DIFF_B_div_Cap_Med,PRCENT_CAP_LOSS_B, M, Cap_Med,Inverse_CapB_l,snr);
end

figure('Name','MSPL2S','NumberTitle','on')
if isempty(F_DIFF_A),
else
[na c_elements_a_mod c_elements_a n_cap_a c_elements_cap_a
AVG_PRCENT_CAP_LOSS_MOD_A]=plot_figures(F_DIFF_A, OptCap_l, DIFF_A,
DIFF_A_div_Cap_Med,PRCENT_CAP_LOSS_A, M, Cap_Med,MSPL2S_CapA_l,snr);
end

figure('Name','MSCL2S','NumberTitle','on')
if isempty(F_DIFF_C),
else
[nc c_elements_c_mod c_elements_c n_cap_c c_elements_cap_c
AVG_PRCENT_CAP_LOSS_MOD_C]=plot_figures(F_DIFF_C, OptCap_l, DIFF_C,
DIFF_C_div_Cap_Med,PRCENT_CAP_LOSS_C, M, Cap_Med,MSCL2S_CapC_l,snr);
end

figure('Name','MSP','NumberTitle','on')
if isempty(F_DIFF_D),
else
[nd c_elements_d_mod c_elements_d n_cap_d c_elements_cap_d
AVG_PRCENT_CAP_LOSS_MOD_D]=plot_figures(F_DIFF_D, OptCap_l, DIFF_D,
DIFF_D_div_Cap_Med,PRCENT_CAP_LOSS_D, M, Cap_Med,MSP_CapD_l,snr);
end

figure('Name','Unordered','NumberTitle','on')
if isempty(F_DIFF_UNORD),
```

```
else
[nu c_elements_u_mod c_elements_u n_cap_u c_elements_cap_u
AVG_PRCENT_CAP_LOSS_MOD_UNORD]=plot_figures(F_DIFF_UNORD, OptCap_l, DIFF_UNORD,
DIFF_UNORD_div_Cap_Med,PRCENT_CAP_LOSS_UNORD, M, Cap_Med,ConvCap_l,snr);
end

%A summary Figure comparing the performance of all the strategies

figure('Name','Summary','NumberTitle','on')
plot_Summary(nu,c_elements_u_mod,c_elements_u,n_cap_u,c_elements_cap_u,AVG_PRCENT_CAP_LOSS_MO
D_UNORD,nb,c_elements_b_mod, na, c_elements_a_mod, nc, c_elements_c_mod, nd,
c_elements_d_mod,c_elements_b,
c_elements_a,c_elements_c,c_elements_d,c_elements_cap_b,c_elements_cap_a,c_elements_cap_c,c_e
lements_cap_d,c_elements_cap_opt,M,snr,F_DIFF_B,F_DIFF_A,F_DIFF_C,F_DIFF_D,F_DIFF_UNORD,AVG_P
RCENT_CAP_LOSS_MOD_B,AVG_PRCENT_CAP_LOSS_MOD_A,AVG_PRCENT_CAP_LOSS_MOD_C,AVG_PRCENT_CAP_LOSS_
MOD_D,n_cap_a,n_cap_b,n_cap_c,n_cap_d,n_cap_opt);

elapsed_time=toc;
elapsed_time=elapsed_time/60;

%Figure showing the empirical CDFs of all ordering strategies

figure('Name','CDFs','NumberTitle','on')
semilogy(n_cap_b,(c_elements_cap_b/M),'-bo',n_cap_a,(c_elements_cap_a/M),'-
ms',n_cap_c,(c_elements_cap_c/M),'-kp',n_cap_d,(c_elements_cap_d/M),'-
gd',n_cap_opt,(c_elements_cap_opt/M),'--k',n_cap_u,(c_elements_cap_u/M),'r');
legend('Inverse','MSPL2S','MSCL2S','MSP','Optimum','Unordered')
title(sprintf('Empirical Capacity CDFs P[C<=R]  SNR = %d dB   Channel realizations %d   3x3
Time: %g min',10*log10(snr),M,elapsed_time));
ylabel('Prob');
xlabel('R')
grid on

%Displaying how many times each ordering attained the maximum capacity

fprintf('\n Number of channel realizations:\t\t\t\t\t\t  %g \n Maximum capacity achieved by
the Inverse ordering: %g \n', [M M-length(F_DIFF_B)])
fprintf('\n Maximum capacity achieved by the MSPL2S ordering: %g \n', M-length(F_DIFF_A))
fprintf('\n Maximum capacity achieved by the MSCL2S ordering: %g \n', M-length(F_DIFF_C))
fprintf('\n Maximum capacity achieved by the MSP ordering: %g \n', M-length(F_DIFF_D))
fprintf('\n Elapsed time: %g \n', elapsed_time)

%Displaying whether the code has been correctly validated

if sum(VAL)== M
    fprintf('\n The code has been validated')
else
    fprintf('\n The code has NOT been validated')
end
```

IRA: Auxiliary functions.

```
function [Bestcut VALIDATION Hopt Bestcut1 CAPConv]=bruta_IRA(n, m, H, snr)
%
%BRUTA_IRA   Obtains all the possible orderings of the channel matrix, finds the capacity of
%            each of them and returns the ordering offering the maximum capacity, the
%            capacity of the MSP ordering, the maximum capacity, the capacity of the
%            unordered detection and validates the code.
%
clear P s O CodVal A B b G VALIDATION un Hopt a; %Clearing variables
P=zeros(1,m);         %Vector to enumerate the columns of the channel matrix.
s=factorial (m);      %# of all possible orderings of the channel matrix
O=zeros(1,s);         %Vector to store the cap. using conventional V-BLAST for all orderings.
CodVal=zeros(1,s);    %Vector to store the product of the after-proc. Sx powers of each
ordering.
Sum_Sx_powersi=zeros(1,s); %Vector to store the sum of the after-proc. Sx power of each
ordering.
G=zeros(n,m);         %Matrix used for the different orderings of the channel.
Hopt=zeros(n,m);      %Matrix to store the optimum ordering.
```

```
Hopt_sum=zeros(n,m); %Matrix to store the ordering that maximizes the after-proc. Sx powers.
A=H;
G=A;

%Obtaining the capacity of the unordered detection

[CValid, CAPConv, sum_Sx_powers]=conventionalV_BLAST(n, m, G, snr);
CAPConv=(fix(CAPConv*1.e4))/1.e4; %Fixing to 4 decimal digits.

%Searching for the optimum ordering and for the one that maximizes the sum of the after-proc.
Sx powers.

clear G sum_Sx_powers CValid
for i=1:m
    P(i)=i;                 %Enumerating the columns of the channel matrix.
end
B=perms(P);                %Obtaining all possible orderings of the columns of the channel
matrix.
for i=1:s
    b=B(i,:);               %Taking the j-th possible ordering.
    for j=1:m
        G(:,j)=A(:,b(j));  %Creating the channel matrix with the j-th ordering
    end
    [CodVal(i) O(i) Sum_Sx_powersi(i)]=conventionalV_BLAST(n, m, G, snr); %Applying
conventional V-BLAST to the j-th ordering, and calculating the capacity; obtaining the sum of
the after-proc. Sx powers of the j-th ordering; validating the code (CodVal(i)).
    clear G b
end
[a ind]=sort(O);            %Ordering the capacities in ascendant order.
Best=a(s);                  %Obtaining the optimum capacity.
k=B(ind(s),:);              %Obtaining the indexes to generate the optimum ordering.
for i=1:m
        Hopt(:,i)=A(:,k(i));      %Obtaining the optimum ordering.
end
Bestcut=(fix(Best*1.e4))/1.e4;    %fixing to 4 decimal digits.

[a ind]=sort(Sum_Sx_powersi);     %Ordering the Sums of the after-proc. Sx powers (MSP) at
each step of all orderings in ascendant order.
Capbestsum=O(ind(s));       %Obtaining the Capacity offered by the MSP ordering.
k=B(ind(s),:);              %Obtaining the indexes to generate the MSP ordering.
for i=1:m
        Hopt_sum(:,i)=A(:,k(i));  %Obtaining the MSP ordering.
end
Bestcut1=(fix(Capbestsum*1.e4))/1.e4;

%Validation of the code

un = unique(CodVal);        %verifying that the prod. of the after-proc. Sx powers are the
same.
VALIDATION=length(un);      %if length(un)==1, all the entries of vector CodVAL are equal,
then the code is validated, else it is not.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cap_Inverse = cap_Inverse_ordering(n, m, H, snr)

% cap_Inverse_ordering   Returns the capacity offered by the Inverse ordering under the IRA.
%

clear C Hord W SxPower Z normas P Znew CAP Capcut svdHord;  %clearing the variables.
C=zeros(1,m);                     %Vector to store the capacity at each step.
Hord = zeros (n,m);               %Matrix to store the ordered channel matrix
SxPower=zeros(1,m);               %Vector to store the after processing signal power at each
step
normas = zeros(1,m);       %Vector to store the norms of the columns of the channels at each
step.
Z=H;                              %Channel matrix to be used for different manipulations.

%Norm of each of the columns of matrix H

for i=1:m
```

```
    normas (i)= norm (Z(:,i));
end

%The last step

[a,ind]=sort(normas);            %Sorting the elements of normas in ascending order
P=eye(n)-Z(:,ind(m))*Z(:,ind(m))'/(norm(Z(:,ind(m)))^2);    %Projection matrix orthogonal to
the column with max. norm
Hord(:,m)=Z(:,ind(m));           %Last column of the ordered channel matrix(the one with max
before-projection norm)
SxPower(m)=(norm(Z(:,ind(m))))^2;%After proc. Sx power at the last step.
Z(:,ind(m))= [];                 %the column with max norm is deleted
Znew=P*Z;             %Projected matrix orthogonal to the column with max before-projection
norm
m=m-1;

%steps from m-1 to 2

for j=1:m-1
    clear normas a P
    for i=1:m-j+1
        normas (i)=norm (Znew(:,i)); %Norm of each of the columns of the projected matrix
    end
[a,ind]=sort(normas);                %Obtaining the position of the column with max norm
P=eye(n)-Znew(:,ind(m-j+1))*Znew(:,ind(m-j+1))'/(norm(Znew(:,ind(m-j+1)))^2); %Projection
matrix orthogonal to the column with max norm at the j-th step (steps from 2 to m-1)
Hord(:,m-j+1)=Z(:,ind(m-j+1));       %Ordering the columns of channel matrix.(the one with
max after-projection norm is selected at each step)
SxPower(m-j+1)=(norm(Znew(:,ind(m-j+1))))^2; %After-proc. signal power at the m-j+1-th step.
Z(:,ind(m-j+1))= [];                 %The column with max norm at the m-j+1-th step is
deleted
Znew(:,ind(m-j+1))= [];              %The column of the projected matrix with max norm is
deleted
Znew=P*Znew;                         %Projection of the projected matrix.
end

%step 1

Hord(:,1)=Z; %The remaining column of the channel matrix is the one allocate at the first
position of the ordered channel matrix
SxPower(1)=(norm(Znew))^2; %After-proc. Sx power at the 1st step.

%obtaining the capacity

m=m+1;
for k=1:m
C(k)=log(1 + snr*(SxPower(k)));    %Capacity at each of the steps.
end
CAP=sum(C);
Cap_Inverse=(fix(CAP*1.e4))/1.e4;  %fixing to 4 decimal digits.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cap_MSCL2S = cap_MSCL2S_ordering (n,m,H,snr)
%
%cap_MSCL2S_ordering:   Returns the capacity offered by the Max Sum Capacity of the Last Two
%                        Steps (MSCL2S) ordering.
%
clear C Hord W SxPower Z normas nrms X Y As P Znew CAP Capcut; %clearing the variables.
C=zeros(1,m);                   %Vector to store the capacities at each step.
Hordc = zeros (n,m);            %Matrix to store the ordered channel matrix
SxPower=zeros(1,m);             %Vector to store the after processing signal power at each
step
normas = zeros(1,m);            %Vector to store the norms of the columns of the channels at
each step.
As=zeros(1,m-1);                %Vector to store the norms of the projected vectors
orthogonal to last column vector in ascendant order.
Ord=zeros(1,m);
Z=H;                            %Channel matrix to be used for different manipulations.

%Last step
```

```
for i=1:m
    normas (i)=norm (Z(:,i));   %Norm of each of the columns of matrix H
end
for j = 1:m
    clear P X Y nrms As
    X = Z;                                      %channel matrix to be reduced by 1 column
    P=eye(n)-X(:,j)*X(:,j)'/(norm(X(:,j))^2); %Projection matrix orthogonal to the column
    X(:,j) = [];                                %deleting j-th column of H
    Y=P*X;                                     %projecting the other columns of H orthogonal to
the selected (j) column
    for i=1:m-1
        nrms(i)=norm(Y(:,i));                   %norms of the projected columns.
    end

    As = sort(nrms);             %ordering the norms of the projected columns in ascendant
order
    Ord(j) = log(1+snr*(As(m-1)^2)) + log(1+snr*(normas(j)^2)); %capacity of the last 2 steps

end
[Ac, ind]=sort(Ord);                            %sorting the max norms of the projected
columns(for each j column)in ascending order. The maximum will give us the last column to
detect.
Hordc(:,m)=Z(:,ind(m));                         %last column of the ordered channel matrix(the
one that maximizes the projected norms)
SxPower(m)=(norm(Z(:,ind(m))))^2;               %After processing signal power at the last
step.
P=eye(n)-Z(:,ind(m))*Z(:,ind(m))'/(norm(Z(:,ind(m)))^2); %Projection matrix orthogonal to the
column
Z(:,ind(m))= [];                                %the column with max norm is deleted
Znew=P*Z;                                       %Projected matrix orthogonal to the column with
max before projection norm

%steps from m-1 to 2

 m=m-1;
for j=1:m-1
    clear normas a P
    for i=1:m-j+1
        normas (i)=norm (Znew(:,i));            %Norm of each of the columns of the projected
matrix
    end
[a,ind]=sort(normas);                           %obtaining the position of the column with max
norm
P=eye(n)-Znew(:,ind(m-j+1))*Znew(:,ind(m-j+1))'/(norm(Znew(:,ind(m-j+1)))^2); %Projection
matrix orthogonal to the column with max norm at the j-th step (steps from 2 to m-1)
Hordc(:,m-j+1)=Z(:,ind(m-j+1));                 %Ordering the columns of channel matrix.(the
one with max after-projection norm is selected at each step)
SxPower(m-j+1)=(norm(Znew(:,ind(m-j+1))))^2;  %After processing signal power at the m-j+1-th
step.
Z(:,ind(m-j+1))= [];                            %The column with max norm at the m-j+1-th step
is deleted
Znew(:,ind(m-j+1))= [];                         %the column of the projected matrix with max
norm is deleted
Znew=P*Znew;                                    %Projection of the projected matrix...
end

%step 1

Hordc(:,1)=Z;                                   %The remaining column of the channel matrix is
the one allocate at the first position of the ordered channel matrix
SxPower(1)=(norm(Znew))^2;                      %After processing signal power at the 1st step.

%Obtaining the capacity

m=m+1;
for k=1:m
C(k)=log(1 + snr*(SxPower(k)));                 %Capacity at each of the steps.
end
CAP=sum(C);
Cap_MSCL2S=(fix(CAP*1.e4))/1.e4;                %Fixing to 4 decimal digits
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cap_MSPL2S = cap_MSPL2S_ordering (n,m,H,snr)
%
%cap_MSPL2S_ordering:      Returns the capacity offered by the Max Sum Powers of the Last 2
%                          Steps (MSPL2S) ordering.
%
clear C Hord W SxPower Z normas nrms X Y As P Znew CAP Capcut;
C=zeros(1,m);             %vector to store the capacities at each step.
Horda = zeros (n,m);      %Matrix to store the ordered channel matrix
SxPower=zeros(1,m);       %vector to store the after processing signal power at each step
normas = zeros(1,m);      %vector to store the norms of the columns of the channels at each
step.
As=zeros(1,m-1);
Ord=zeros(1,m);
Z=H;                      %Channel matrix to be used for different manipulations.

%Norm of each of the columns of the matrix H

for i=1:m
    normas (i)=norm (Z(:,i));
end
for j = 1:m
    clear P X Y nrms As
    X = Z;                                   % channel matrix to be reduced by 1 column
    P=eye(n)-X(:,j)*X(:,j)'/(norm(X(:,j))^2);%Projection matrix orthogonal to the column
    X(:,j) = [];                             %deleting j-th column of H
    Y=P*X;                                   %projecting the other columns of H orthogonal to
the selected (j) column
    for i=1:m-1
        nrms(i)=norm(Y(:,i));                %norms of the projected columns.
    end
    As = sort(nrms);                         %sorting the norms of the projected columns in
ascending order in order to take the maximum.
    Ord(j) = (As(m-1))^2+ (normas(j))^2;     %taking the maximum norm of the projected
columns (projected orthogonal to the selected (j) column)and adding norm of the j-th column
end
[Ac, ind]=sort(Ord);                         %sorting the max norms of the projected
columns(for each j column)in ascending order.
                                             %The maximum will give us the last column to
detect.
Horda(:,m)=Z(:,ind(m));                      %last column of the ordered channel matrix(the
one that maximizes the projected norms)
SxPower(m)=(norm(Z(:,ind(m))))^2;            %After processing signal power at the last step.
P=eye(n)-Z(:,ind(m))*Z(:,ind(m))'/(norm(Z(:,ind(m)))^2);              %Projection matrix
orthogonal to the column
Z(:,ind(m))= [];                             %the column with max norm is deleted
Znew=P*Z;                                    %Projected matrix in the direction orthogonal to
the column with with max before projection norm
m=m-1;

%steps from m-1 to 2

for j=1:m-1
    clear normas a P
    for i=1:m-j+1
        normas (i)=norm (Znew(:,i));         %Norm of each of the columns of the projected
matrix
    end
[a,ind]=sort(normas);                        %obtaining the position of the column with max
norm
P=eye(n)-Znew(:,ind(m-j+1))*Znew(:,ind(m-j+1))'/(norm(Znew(:,ind(m-j+1)))^2); %Projection
matrix orthogonal to the column with max norm at the j-th step (steps from 2 to m-1)
Horda(:,m-j+1)=Z(:,ind(m-j+1));              %Ordering the columns of channel matrix.(the one
with max after-projection norm is selected at each step)
SxPower(m-j+1)=(norm(Znew(:,ind(m-j+1))))^2; %After processing signal power at the m-j+1-th
step.
Z(:,ind(m-j+1))= [];                         %The column with max norm at the m-j+1-th step
is deleted
```

```
Znew(:,ind(m-j+1))= [];                          %the column of the projected matrix with max
norm is deleted
Znew=P*Znew;                                     %Projection of the projected matrix...
end

%step 1

Horda(:,1)=Z;                                   %The remaining column of the channel matrix is the
one allocated at the first position of the ordered channel matrix
SxPower(1)=(norm(Znew))^2;                       %After processing signal power at the 1st step.

m=m+1;
for k=1:m
C(k)=log(1 + snr*(SxPower(k)));                  %Capacity at each of the steps.
end
CAP=sum(C);
Cap_MSPL2S=(fix(CAP*1.e4))/1.e4;                 %fixing to 4 decimal digits

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [CValid, CAPConv, sum_Sx_powers] = conventionalV_BLAST(n, m, G, snr)

% CONVENTIONALV_BLAST   Returns the capacity of a system using conventional (without
%                       ordering) V-BLAST under the IRA for a given ordering. Also gives the
%                       sum and the product of the after processing Sx powers at each step
%
clear Aw CapConv Conv CValid CV Q P C;  %clearing the variables.
Pow=zeros(1,m);               %Row vector to allocate the after processing signal power at
each step
X = G;                        %Channel matrix to be used for different manipulations.
for i = 1:m-1                 %Detection Process, i-th step
    clear Q P;
    X(:,1) = [];              %deleting the first caolumn column of H
    Q = null(X');             %orthonormal basic of the null space of X'
    P = Q*Q';                 %orthogonal projection matrix
    wj(:,i) = P*G(:,i);       %orthogonal projection of i-th column
    Pow(i) = norm(wj(:,i))^2; %after-proc. Sx power at the i-th step
end
wj(:,m) = G(:,m);             %it is not necessary to project out more interference, it is
the last column.
Pow(m) = norm(G(:,m))^2;      %after-proc. Sx power at the last (m) step
CV=prod(Pow);                 %product of the after-proc. Sx powers at diferent steps.
CValid =(fix(CV*1.e4))/1.e4;       %fixing to 4 decimal digits

%Obtaining the capacity

for j=1:m
C(j)=log(1 + snr*(Pow(j)));
end
CAPConv=sum(C);               %Total capacity
sum_Sx_powers = sum(Pow);     %Sum of the after-proc Sx powers

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [nb c_elements_b_mod c_elements_b n_cap_b c_elements_cap_b
AVG_PRCENT_CAP_LOSS_MOD_B] = plot_figures(F_DIFF_B, OptCap_l, DIFF_B,
DIFF_B_div_Cap_Med,PRCENT_CAP_LOSS_B, M, Cap_Med,Anti_Foschini_CapB_l, snr)
%
%plot_figures   Plots graphs showing the performance of the specific ordering
%
clear nb n_elements_b_mod c_elements_b_mod n_elements_b c_elements_b n_cap_b n_elements_cap_b
c_elements_cap_b AVG_PRCENT_CAP_LOSS_MOD_B len_mod id_n_elements_b_mod id_c_elements_b_mod
len id_n_elements_b id_c_elements_b;
%
%
for i=1:length(F_DIFF_B)
OptCap_l_mod_b(i)=OptCap_l(F_DIFF_B(i));%obtaining the optimum capacity for the cases where
the specific ordering did not achieve the maximum capacity
DIFF_B_mod(i)=DIFF_B(F_DIFF_B(i));      %collecting the difference between the maximum cap
and the capacity offered by the specific ordering
```

```
DIFF_B_div_Cap_Med_MOD(i)=DIFF_B_div_Cap_Med(F_DIFF_B(i));%normalizing the difference by the
mean capacity
PRCENT_CAP_LOSS_MOD_B(i)=PRCENT_CAP_LOSS_B(F_DIFF_B(i)); %expressing the difference between
the optimum cap and the capacity offered by the specific ordering in percent
end
AVG_PRCENT_CAP_LOSS_MOD_B=sum(PRCENT_CAP_LOSS_MOD_B)/length(PRCENT_CAP_LOSS_MOD_B);%Average
of the loss in capacity (in percent)

nb=[0:(max(DIFF_B_div_Cap_Med)/50):max(DIFF_B_div_Cap_Med)];
n_elements_b_mod = hist(DIFF_B_div_Cap_Med_MOD,nb); %obtaining the histogram of DELTA CAP
(normalized by the Mean Cap) of the cases where the maximum capacity was not achieved
c_elements_b_mod = cumsum(n_elements_b_mod);        %Obtaining the cumulative sum

%plotting the histogram of DELTA CAP (normalized by the Mean Cap. and by the total channel
realizations)of the cases where the maximum capacity was not achieved.

subplot(3,2,1);
bar(nb,(n_elements_b_mod/M));
title('    Histogram of DeltaCap/MeanCapOpt (Channels with DeltaCap>0)');
ylabel(sprintf('# of channels\n____________________\ntotal # of realizations'));
xlabel('DeltaCap/MeanCapOpt');

%Plotting the cumulative sum (normalized by the total channel realizations)

subplot(3,2,2);
len_mod = length(n_elements_b_mod);
id_n_elements_b_mod= len_mod:-1:1;
n_elements_b_mod=n_elements_b_mod(id_n_elements_b_mod);
c_elements_b_mod=cumsum(n_elements_b_mod);
len_mod=length(c_elements_b_mod);
id_c_elements_b_mod = len_mod:-1:1;
c_elements_b_mod=c_elements_b_mod(id_c_elements_b_mod);
semilogy(nb,(c_elements_b_mod/M));
title('Empirical P[DeltaCap/MeanCapOpt>=x] (Channels with DeltaCap>0)');
ylabel('Prob');
xlabel('x');
grid on;

n_elements_b = hist(DIFF_B_div_Cap_Med,nb);    %obtaining the histogram of DELTA CAP
(normalized by the Mean Cap) of all the orderings(including capacity achieving and no
capacity achieving)
c_elements_b = cumsum(n_elements_b);%Obtaining the cumulative sum

%plottimg the histogram of DELTA CAP (normalized by the Mean Cap and by the total channel
realizations) of all cases.

subplot(3,2,3);
bar(nb,(n_elements_b/M));
title('Histogram of DeltaCap/MeanCapOpt (All Channels)');
ylabel(sprintf('# of channels\n____________________\ntotal # of realizations'));
xlabel('DeltaCap/MeanCapOpt');

%Plotting the cumulative sum (normalized by the total channel realizations)

subplot(3,2,4);
len = length(n_elements_b);
id_n_elements_b= len:-1:1;
n_elements_b=n_elements_b(id_n_elements_b);
c_elements_b=cumsum(n_elements_b);
len=length(c_elements_b);
id_c_elements_b = len:-1:1;
c_elements_b=c_elements_b(id_c_elements_b);
semilogy(nb,(c_elements_b/M));
title('Empirical P[DeltaCap/MeanCapOpt>=x] (All Channels)');
ylabel('Prob');
xlabel('x');
grid on;

n_cap_b=[0:(max(Anti_Foschini_CapB_l)/50):max(Anti_Foschini_CapB_l)];
n_elements_cap_b=hist(Anti_Foschini_CapB_l,n_cap_b);
c_elements_cap_b=cumsum(n_elements_cap_b);
```

```
%plottimg the histogram of the capacity achieved by the specific ordering (normalized by the
total channel realizations).

subplot(3,2,5);
hold on;
bar(n_cap_b,(n_elements_cap_b/M),'FaceColor', 'r');
plot(0*(n_elements_cap_b/M)+Cap_Med,(n_elements_cap_b/M));
title('Histogram of Capacity');
ylabel(sprintf('# of channels\n____________________\ntotal # of realizations'));
xlabel(sprintf('Capacity (Bits/Sec/Hz)\n MeanCapOpt = %.2g Bits/Sec/Hz    SNR = %d dB
Channel realizations %d',Cap_Med,10*log10(snr),M));
hold off;

%Plotting the CDF of the specific ordering

subplot(3,2,6);
semilogy(n_cap_b,(c_elements_cap_b/M));
title('Empirical CDF: P[C<=R]');
ylabel('Prob');
xlabel('R');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function[] = plot_Summary(nu, c_elements_u_mod, c_elements_u, n_cap_u, c_elements_cap_u,
AVG_PRCENT_CAP_LOSS_MOD_UNORD, nb, c_elements_b_mod, na, c_elements_a_mod, nc,
c_elements_c_mod, nd, c_elements_d_mod,c_elements_b, c_elements_a, c_elements_c,
c_elements_d, c_elements_cap_b, c_elements_cap_a, c_elements_cap_c, c_elements_cap_d,
c_elements_cap_opt, M, snr, F_DIFF_B, F_DIFF_A, F_DIFF_C, F_DIFF_Max_sum_P, F_DIFF_UNORD,
AVG_PRCENT_CAP_LOSS_MOD_B, AVG_PRCENT_CAP_LOSS_MOD_A, AVG_PRCENT_CAP_LOSS_MOD_C,
AVG_PRCENT_CAP_LOSS_MOD_Max_sum_P, n_cap_a, n_cap_b, n_cap_c, n_cap_d, n_cap_opt)

%plot_Summary  Plots 3 graphs comparing all the studied orderings and show a table showing
%              more details of the comparison regarding the capacity.
%
%plottimg the cumulative sum (for all orderings)of DELTA CAP (normalized by the Mean Cap. and
by %the total channel realizations)of the cases where the maximum capacity was not achieved.
%
subplot(2,2,1)
semilogy(nb,(c_elements_b_mod/M),'b',na,(c_elements_a_mod/M),'r',nc,(c_elements_c_mod/M),'k',
nd,(c_elements_d_mod/M),'g',nu,(c_elements_u_mod/M),'y')
legend('Inverse','MSPL2S','MSCL2S','MSP','Unordered')
title(sprintf('Empirical P[DeltaCap/MeanCapOpt>=x]\n All strategies (Channels with
DeltaCap>0)'));
ylabel('Prob');
xlabel('x')
grid on

%plottingg the cumulative sum (for all orderings)of DELTA CAP (normalized by the Mean Cap.
and by %the total channel realizations)of all cases

subplot(2,2,2)
semilogy(nb,(c_elements_b/M),'b',na,(c_elements_a/M),'r',nc,(c_elements_c/M),'k',nd,(c_elemen
ts_d/M),'g',nu,(c_elements_u/M),'y')
legend('Inverse','MSPL2S','MSCL2S','MSP','Unordered')
title(sprintf('Empirical P[DeltaCap/MeanCapOpt>=x]\n All strategies (All Channels)'));
ylabel('Prob');
xlabel('x')
grid on

%Plotting the CDFs of all orderings (normalized by the total channel realizations)

subplot(2,2,3)
semilogy(n_cap_b,(c_elements_cap_b/M),'b',n_cap_a,(c_elements_cap_a/M),'r',n_cap_c,(c_element
s_cap_c/M),'k',n_cap_d,(c_elements_cap_d/M),'g',n_cap_opt,(c_elements_cap_opt/M),'k',n_cap_u,
(c_elements_cap_u/M),'y')
legend('Inverse','MSPL2S','MSCL2S','MSP','Optimum','Unordered')
title('Empirical Capacity CDFs P[C<=R] (All strategies)');
ylabel('Prob');
xlabel('R')
```

```
grid on

%Generating the summary table
  %Creating an invisible plot, to create the space where the table will be generated.
x=[0:12]
y=x
subplot(2,2,4)
plot(x,y,'w')

  %Summary table

title(sprintf('SNR = %d dB   Channel realizations %d',10*log10(snr),M))

text(0,8,'   Inverse')
text(0,7,'   MSPL2S')
text(0,6,'   MSCL2S')
text(0,5,'   MSP')
text(0,4,'   Unordered')

text(4.5,9.5,'                                    AVG')
text(4.5,9,'Not Opt      Percent     Cap Loss')
text(4.7,8,['', num2str(length(F_DIFF_B)),'
',num2str((length(F_DIFF_B)/M)*100,2),'%','
',num2str(AVG_PRCENT_CAP_LOSS_MOD_B,2),'%'])
text(4.7,7,['', num2str(length(F_DIFF_A)),'
',num2str((length(F_DIFF_A)/M)*100,2),'%','
',num2str(AVG_PRCENT_CAP_LOSS_MOD_A,2),'%'])
text(4.7,6,['', num2str(length(F_DIFF_C)),'
',num2str((length(F_DIFF_C)/M)*100,2),'%','
',num2str(AVG_PRCENT_CAP_LOSS_MOD_C,2),'%'])
text(4.7,5,['', num2str(length(F_DIFF_Max_sum_P)),'
',num2str((length(F_DIFF_Max_sum_P)/M)*100,2),'%','
',num2str(AVG_PRCENT_CAP_LOSS_MOD_Max_sum_P,2),'%'])
text(4.7,4,['', num2str(length(F_DIFF_UNORD)),'
',num2str((length(F_DIFF_UNORD)/M)*100,2),'%','
',num2str(AVG_PRCENT_CAP_LOSS_MOD_UNORD,2),'%'])
```

IRA: Combinatorial code to find N (maximum number of orderings that can satisfy the necessary optimality conditions)

```
function Descartes2

%Descartes2:   Displays the maximum number of combinations that can meet the necessary
%              optimality conditions based on the fact that from 2 combinations with a
%              swapped pair, only one can survive (the one with the highest number to the
%              right).Combinatorial code, based only on the indexes of the columns, not on
%              the projections.
%

m=3;           %number of columns of the channel matrix
s=factorial(m);%number of possible combinations
col=[1:m];     %enumerating the number of columns
A=perms(col);  %All possible combinations of columns
index=1;       %Index to store the columns that will not be eliminated.
a=0;

%First necessary optimality condition

for i=1:s
    b=A(i,:);
    if b(1)>b(2);
        X(index,:)=b;
        index=index+1;
    end
end

%Eliminating all duplicates rows from the storage matrix and ordering the rows of the matrix
%favorably for the next conditions
```

```
X = unique(X,'rows','first');

%rest of the necessary conditions (except the last one)

clear A b c index
index=1;                    %Index to store the columns that will not be eliminated.
A=X;                        %Surviving combinations after the 1st necessary cond.
H=A;                        %Matrix for different manipulation
for i=1:m-3
    clear X;
    ind=[1:i];              %indices of the entries to the left of the swapped pair
    ond=[i+3:m];            %indices of the entries to the right of the swapped pair

    for k=1:(size(A,1)-1)
        b=A(k,:);           %Taking the k-th combination
        H(1,:)=[];          %Eliminating the k-th combination from the rest of comb.

        %Comparing the rest of combinations with the k-th selected.

        for j=1:size(H,1)
            c=H(j,:);       %Taking the j-th combination to compare it with the k-th selected
            bmod=[b(ind) b(ond)]; %Creating a vector to compare both sides of the swapped
pairs.
            cmod=[c(ind) c(ond)]; %Creating a vector to compare both sides of the swapped
pairs.

            %Comparing both sides of the swapped pairs, if they are equal the combination
that
            %has the greater number of the swapped pair to the left will be selected (at the
end
            %we will reverse the orderings and the greater number of the swapped pair will be
to
            %the right)

            if bmod==cmod   %Comparing both sides of the swapped pairs.

                 if b(i+1)> b(i+2);
                    X(index,:)=b;
                    index=index+1;
                    a=1;
                    break
                 else X(index,:)=c;
                      index=index+1;
                      a=1;
                      break
                 end
            end
        end

        %if there is no combination with a swapped pair (as compared to the k-th), the k-th
will
        %survive.

        if a==1
          else
             X(index,:)=b;
             index=index+1;
             a=0;
        end
        a=0;
    end

    %Taking also the last ordering, just in case it wasn't taken during the
    %comparison. If it was already taken, it will go away in the next step.

    X(index,:)=H;

    %Eliminating all duplicates rows from the storage matrix and ordering the rows of the
matrix
    %favorably for the next conditions
```

```
    X = unique(X,'rows','first');
    A=X;                        %Surviving combinations after the i-th necessary cond.
    H=A;                        %Matrix for different manipulation
    index=1;
end

%The last necessary condition

ind=[1:m-2];
    for k=1:(size(A,1)-1)
        b=A(k,:);           %Taking the k-th combination
        H(1,:)=[];          %Eliminating the k-th combination from the rest of comb.

        %Comparing the rest of the combinations with the k-th.

        for j=1:size(H,1);
            c=H(j,:);
             if b(ind)==c(ind)
                 if b(m-1)> b(m)
                    X(index,:)=b;
                    index=index+1;
                    a=1;
                 else X(index,:)=c;
                      index=index+1;
                      a=1;
                 end
             end
        end

        %if there is no combination with a swapped pair (as
        %compared to the k-th), the k-th will survive.

        if a==1;
        else
        X(index,:)=b;
        index=index+1;
        a=0;
        end
        a=0;
    end

%Taking also the last ordering, just in case it wasn't taken during the comparison. If it was
%already taken, it will go away in the next step.

X(index,:)=H;

%Eliminating all duplicates rows from the storage matrix and showing the number of
combinations %remaining.

X = unique(X,'rows','first');
x=size(X,1)

%inverting the order of the columns in the storage matrix, then the remaining combinations
are in %accordance with the necessary conditions.

id_n_elements_c= m:-1:1;
X=X(:,id_n_elements_c)
```

IPRA: Principal function.

```
function principalFWF

%principal.m   Returns the CDF graphs of the Inverse ordering, the optimum ordering and the
%              unordered detection under IRA, WF and FWF. Returns the empirical histogram
%              showing the probability that the Inverse ordering under FWF offers the same
%              capacity as the Inverse ordering under WF.
%              Evaluates if the optimum ordering under the WF is the same as the optimum
%              ordering under the FWF.
%
```

```
%Inicialization data

SNR_dB=20;
m=3;                                %number of transmit antennas.
n=3;                                %number of receive antennas.
Pt=m;                               %total power constraint.
M=100;                              %number of channel realizations.
snr=10^(SNR_dB/10);                 %Received SNR.10^(dB/10).
Cap_Unord_fractional_WF_l=zeros(1,M);%Vector to store the capacity of the unordered detection
for each channel realization under the FWF.
Cap_Inverse_WF_l=zeros(1,M);        %Vector to store the capacity of the Inverse ordering
for each channel realization under the WF.
Opt_Cap_WF_l=zeros(1,M);            %Vector to store the capacity of the optimum ordering
for each channel realization under the WF.
Opt_Cap_fractional_WF_l=zeros(1,M);  %Vector to store the capacity of the optimum ordering
for each channel realization under the FWF.
Cap_Inverse_fractional_WF_l=zeros(1,M); %Vector to store the capacity of the Inverse ordering
for each channel realization under the FWF.
Hopts_WF_fractional_WF_equal_l=zeros(1,M); %Vector to follow the # of times that the optimum
ord. under the WF is the same as the opt. ord. under the FWF.
tic;
fprintf('\nPlease wait, Matlab is running the simulation ...\n')

%generating M channel realizations and saving the capacity offered by each
%ordering strategy

for l=1:M
    clear Cap_Unord_fractional_WF Opt_Cap_WF Cap_Inverse_WF Opt_Cap_fractional_WF
Cap_Inverse_fractional_WF Hopts_WF_fractional_WF_equal;
    H = 1/sqrt(2)*(randn(n,m) + 1i*randn(n,m)); %Random channel matrix, with entries iid CN-
(0,1).

    %Obtaining the capacity of each ordering for the given channel realization

    [Cap_Unord_fractional_WF Opt_Cap_WF Cap_Inverse_WF Opt_Cap_fractional_WF
Cap_Inverse_fractional_WF Hopts_WF_fractional_WF_equal] = brutaFWF(n, m, H, snr,Pt);

    %Saving the capacity of each ordering for the given channel realization

    Cap_Unord_fractional_WF_l(l)=Cap_Unord_fractional_WF;
    Cap_Inverse_WF_l(l)=Cap_Inverse_WF;
    Opt_Cap_WF_l(l)=Opt_Cap_WF;
    Opt_Cap_fractional_WF_l(l)=Opt_Cap_fractional_WF;
    Cap_Inverse_fractional_WF_l(l)=Cap_Inverse_fractional_WF;

    Hopts_WF_fractional_WF_equal_l(l)=Hopts_WF_fractional_WF_equal; %Checking if the opt.
ord. under the WF and under the FWF are the same.
end

%Obtaining the number of bins and the cumulative sum for the capacity achieved by each
ordering of interest.

[n_Cap_Unord_fractional_WF c_elements_Cap_Unord_fractional_WF] =
cumulative_elements(Cap_Unord_fractional_WF_l);
[n_Opt_Cap_WF c_elements_Opt_Cap_WF] = cumulative_elements(Opt_Cap_WF_l);
[n_Cap_Inverse_WF c_elements_Cap_Inverse_WF] = cumulative_elements(Cap_Inverse_WF_l);
[n_Opt_Cap_fractional_WF c_elements_Opt_Cap_fractional_WF] =
cumulative_elements(Opt_Cap_fractional_WF_l);
[n_Cap_Inverse_fractional_WF c_elements_Cap_Inverse_fractional_WF] =
cumulative_elements(Cap_Inverse_fractional_WF_l);


elapsed_time=toc;
elapsed_time=elapsed_time/60

%Plotting the CDF graphs

figure(1),
semilogy(n_Cap_Unord_fractional_WF,(c_elements_Cap_Unord_fractional_WF/M),'-bo',
n_Opt_Cap_WF,(c_elements_Opt_Cap_WF/M),'-k+',
n_Opt_Cap_fractional_WF,(c_elements_Opt_Cap_fractional_WF/M));
```

```
legend('Unordered FWF','Opt WF','Opt fractional WF')
title(sprintf('Empirical Capacity CDFs P[C<=R]  SNR = %d dB   Channel realizations %d   %dx%d
Time: %g min',10*log10(snr),M,m,n,elapsed_time));
ylabel('Prob');
xlabel('R')
grid on

figure(2),
semilogy(n_Cap_Inverse_WF,(c_elements_Cap_Inverse_WF/M),'-
rd',n_Cap_Inverse_fractional_WF,(c_elements_Cap_Inverse_fractional_WF/M),'-k+',
n_Opt_Cap_fractional_WF,(c_elements_Opt_Cap_fractional_WF/M));
legend('Inverse WF','Inverse FWF','Opt FWF')
title(sprintf('Empirical Capacity CDFs P[C<=R]  SNR = %d dB   Channel realizations %d   %dx%d
Time: %g min',10*log10(snr),M,m,n,elapsed_time));
ylabel('Prob');
xlabel('R')
grid on

%Plotting the histogram showing the prob. that the Inverse ordering achieves the same
capacity under the WF and under the FWF.

Diff_Inversees = Cap_Inverse_fractional_WF_l - Cap_Inverse_WF_l;%vector with the difference
in capacity between the Inverse ordering under FWF and under WF.
max_difference=max(Diff_Inversees);%Maximum difference in capacity between the Inverse
ordering under FWF and under WF
Cap_med_Inverse=sum(Cap_Inverse_fractional_WF_l)/M;%Mean capacity offered by Inverse ordering
under the FWF

histo_Diff_Inversees = hist(Diff_Inversees,[0 0.001]);%histogram of Diff_Inversees. To see
the empirical prob. that both power allocation strategies end up with equal cap.

figure (3),
bar([0 0.001],(histo_Diff_Inversees/M));%Plotting previous histogram but the y axis divided
by the total channel realizations to express it in term of prob. The bin at 0 is
corresponding to both strategies achieving equal cap. The bin at 0.001 is corresponding to
both strategies achieving different cap.
axis([-0.001 0.002 0 1])
title(sprintf('Prob. dist. of the diff between Cap. of the Inverse ord. under the FWF and
under the WF. SNR = %d dB   Ch. Realiz. %d   %dx%d',10*log10(snr),M,m,n));
ylabel('Prob');
xlabel(sprintf('0=equal cap      0.001=different cap       Max diff in Cap: %.3g       Cap.
media: %.3g',max_difference,Cap_med_Inverse))

%Displaying the number of times that the optimum orderings under the WF and the FWF are the
same.

if sum(Hopts_WF_fractional_WF_equal_l)==M
    fprintf('\n The optimum ordering under the WF is always equal to the optimum ordering
under the FWF \n')
else
    fprintf('\n The optimum ordering under the WF was equal to the optimum ordering under the
FWF %g times \n',sum(Hopts_WF_fractional_WF_equal_l))
end
```

IPRA: Auxiliary functions.

```
function [Cap_Unord_fractional_WF Opt_Cap_WF Cap_Inverse_WF Opt_Cap_fractional_WF
Cap_Inverse_fractional_WF Hopts_WF_fractional_WF_equal] = brutaFWF(n, m, H, snr,Pt)
%
%BRUTA   Returns the following values
%        Capacity of the unordered V-Blast under the FWF.
%        Capacity of the optimum ordering under the WF.
%        Capacity of the optimum ordering under the FWF.
%        Capacity of the Inverse ordering under the WF.
%        Capacity of the Inverse ordering under the FWF.
%        Also compares the optimum ordering under the WF with the optimum ordering under the
%        FWF and if they are equal returns Hopts_WF_fractional_WF_equal=1.
%
```

```
clear P s A G B b a2 index index2 Inverse WF Cap_Unord_fractional_WF
clear Opt_Cap_IRA Opt_Cap_WF Cap_Inverse_IRA Cap_Inverse_WF a3 index3 Opt_Cap_fractional_WF ;
clear max_cap_index_WF max_cap_index_fractional_WF equals Hopts_WF_fractional_WF_equal

s=factorial (m);              %# of all possible orderings of the channel matrix
WF=zeros(1,s);                %vector to store the capacity of each ordering under the WF
fractional_WF=zeros(1,s);     %vector to store the capacity of each ordering under the FWF
G=zeros(n,m);                 %Matrix used for the different orderings of the channel.
A=H;
P=[1:m];                      %Enumerating the columns of the channel matrix.

%Obtaining the unordered capacity under the FWF, and the optimum capacities under the WF and
under the FWF

B=perms(P);                   %all possible orderings of the columns of the channel matrix.
for i=1:s
    b=B(i,:);                 %Taking the i-th possible ordering.
    G=A(:,b);                 %Creating the channel matrix with the i-th ordering

    fractional_WF(i)=fractional_water_filling(G, m, snr,Pt);%Obtaining Cap of the i-th
ordering under the FWF.
    WF(i)= water_filling( G, m, snr,Pt);     %Obtaining Cap of i-th ordering under the WF.
    clear G b
end

Cap_Unord_fractional_WF = fractional_WF(1);  %Capacity of the unordered detection under the
FWF.

[a2 index2]=sort(WF);          %Ordering the capacities under the WF in ascendant order and
obtaining the indexes.
Opt_Cap_WF=a2(s);              %Obtaining the optimum capacity under the WF

[a3 index3]=sort(fractional_WF);%Ordering the capacities under the FWF in ascendant order and
obtaining the indexes.
Opt_Cap_fractional_WF=a3(s);    %Obtaining the optimum capacity under the FWF

%Evaluating the Inverse ordering under the WF and under the FWF

Inverse=Inverse_ordering(n, m, H);              %Obtaining the inverse ordering
Cap_Inverse_WF=water_filling( Inverse,m,snr,Pt);%Capacity of the Inverse ordering under the
WF.
Cap_Inverse_fractional_WF=fractional_water_filling(Inverse, m, snr,Pt);%Capacity of the
Inverse ordering under the FWF.

%Checking if the optimum orderings under the WF and under the FWF are the same

max_cap_index_WF = max_cap_index(WF, s);%indexes of the orderings that achieve maximum
capacity under the WF
max_cap_index_fractional_WF = max_cap_index(fractional_WF, s);%indexes of the orderings that
achieve maximum capacity under the FWF

equals = compare_max_cap_index (max_cap_index_WF, max_cap_index_fractional_WF );%Finding if
there is at least one ordering that is optimum under WF and under FWF at the same time.

if equals
    Hopts_WF_fractional_WF_equal =1;
else
    Hopts_WF_fractional_WF_equal =0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cap_Fractional_WF = fractional_water_filling(G, m, snr,Pt)

% fractional_water_filling   Returns the capacity under the FWF algorithm.
%
clear r1 r M

r1=(2^m)/2;       %Number of rows to eliminate from M (All rows where the first column is
active)
r=[1:r1]';          %Vector enumerating the rows to eliminate from H.
```

```
M = combn([0 1],m); %All possible combinations of active columns.
M(r,:)=[];          %Eliminating all rows where the first column is active

for i=1:r1
    clear f m1 X
    f=find(M(i,:)); %Finding the indexes of the active columns in the i-th combination of
active columns.
    m1=length(f);   %number of active columns in the i-th combination of active columns.(will
be the m used to call the function water filling)
    X=G(:,f);       %New channel matrix formed by the active columns.
    Cap(i) = water_filling(X, m1, snr,Pt);%Applying water filling to the channel matrix
formed by the active columns
end
Cap_Fractional_WF=sort(Cap);              %Ordering the capacities in ascendant order
Cap_Fractional_WF=Cap_Fractional_WF(r1);  %Taking the maximum capacity

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cap = water_filling(G, m, snr,Pt)
%
%water_filling   Returns the capacity under the WF algorithm.

clear Pow a
Pow = V_BLAST_Channel_Gains(m, G);
a=sort(Pow);
for j=m:-1:1
    clear mio1 mio2 mio miu_m p p_g_snr Cap

mio1 = Pt/j;
mio2 = 1./a;
mio = (1/(j*snr))*sum(mio2);
miu_m = mio1+mio;           %Water level for m active streams.

if miu_m >1/(snr*a(1))
    mio2=mio2*(1/snr);
    p=miu_m - mio2;         %vector storing the powers associated to each stream.
    p_g_snr = 1 + (p.*a*snr);
    Cap = log(p_g_snr);     %capacity of each stream
    Cap=sum(Cap);           %Total capacity
    break
else
    a(1)=[];                %turning off the minimum stream
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Pow = V_BLAST_Channel_Gains( m, G)

% V_BLAST_Channel_Gains   Returns the channel gains of a system using conventional (without
%                         ordering) V-BLAST.
%
clear wj Q P C  Pow;          %clearing the variables.

Pow=zeros(1,m);               %vector to allocate the after processing signal power at each
step.
X = G;                        %channel matrix to be used for different manipulations.
for i = 1:m-1                 %detection Process, i-th step
    clear Q P;
    X(:,1) = [];              %deleting j-th column of H
    Q = null(X');             %orthonormal basic of the null space of X'
    P = Q*Q';                 %orthogonal projection matrix
    wj(:,i) = P*G(:,i);       %orthogonal projection of i-th column of H
    Pow(i) = norm(wj(:,i))^2; %after processing Sx power at the i-th step
end

wj(:,m) = G(:,m);             %the last column
Pow(m) = norm(G(:,m))^2;      %after processing Sx power at the last (m) step

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [M,IND] = combn(V,N)
% COMBN - all combinations of elements
%   M = COMBN(V,N) returns all combinations of N elements of the elements in
%   vector V. M has the size (length(V).^N)-by-N.
%
%   [M,I] = COMBN(V,N) also returns the index matrix I so that M = V(I).
%
%   V can be an array of numbers, cells or strings.
%
error(nargchk(2,2,nargin)) ;

if isempty(V) || N == 0,
    M = [] ;
    IND = [] ;
elseif fix(N) ~= N || N < 1 || numel(N) ~= 1 ;
    error('combn:negativeN','Second argument should be a positive integer') ;
elseif N==1,
    % return column vectors
    M = V(:) ;
    IND = (1:numel(V)).' ;
else
    % speed depends on the number of output arguments
    if nargout<2,
        M = local_allcomb(V,N) ;
    else
        % indices requested
        IND = local_allcomb(1:numel(V),N) ;
        M = V(IND) ;
    end
end

% LOCAL FUNCTIONS

function Y = local_allcomb(X,N)
if N>1
    % create a list of all possible combinations of N elements
    [Y{N:-1:1}] = ndgrid(X) ;
    % concatenate into one matrix, reshape into 2D and flip columns
    Y = reshape(cat(N+1,Y{:}),[],N) ;
else
    % no combinations have to be made
    Y = X(:) ;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Hord = Inverse_ordering(n, m, H)

% Inverse_ORDERING   Orders the channel matrix in a specific way. The stream with highest % %
%                    after-processing channel gain is detected last.
%

clear C Hord W SxPower Z normas P Znew CAP Capcut svdHord; %clearing the variables.
C=zeros(1,m);                   %vector to store the capacities at each step.
Hord = zeros (n,m);             %Matrix to store the ordered channel matrix
normas = zeros(1,m);            %vector to store the norms of the columns of the channels at
each step.
Z=H;                            %Channel matrix to be used for different manipulations.

%step m

for i=1:m
    normas (i)=norm (Z(:,i));   %Norm of each of the columns of matrix H
end

[a,ind]=sort(normas);           %sorting the elements of normas in ascending order
P=eye(n)-Z(:,ind(m))*Z(:,ind(m))'/(norm(Z(:,ind(m)))^2);    %Projection matrix orthogonal to
the column with max norm
Hord(:,m)=Z(:,ind(m));          %last column of the ordered channel matrix(the one with max
before-projection norm)
Z(:,ind(m))= [];                %the column with max norm is deleted
```

```
Znew=P*Z;                       %Projected matrix in the direction orthogonal to the column
with with max before projection norm

%steps from m-1 to 2

m=m-1;
for j=1:m-1
    clear normas a P
    for i=1:m-j+1
        normas (i)=norm (Znew(:,i)); %Norms of each of the columns of the projected matrix
    end
[a,ind]=sort(normas);           %obtaining the position of the column with max norm
P=eye(n)-Znew(:,ind(m-j+1))*Znew(:,ind(m-j+1))'/(norm(Znew(:,ind(m-j+1)))^2); %Projection
matrix orthogonal to the column with max norm at the j-th step (steps from 2 to m-1)
Hord(:,m-j+1)=Z(:,ind(m-j+1));  %Ordering the columns of channel matrix.(the one with max
after-projection norm is selected at each step)
Z(:,ind(m-j+1))= [];            %The column with max norm at the m-j+1-th step is deleted
Znew(:,ind(m-j+1))= [];         %the column of the projected matrix with max norm is deleted
Znew=P*Znew;                    %Projection of the projected matrix...
end

%step 1

Hord(:,1)=Z;                    %The remaining column is the first

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function out = compare_max_cap_index (max_cap_index_1, max_cap_index_2 )
%
% compare_max_cap_index   Finds if two vectors have at least one equal entry. If they have at
%                         least one equal entry out= the first entry that is common %in both
%                         vectors. Else out=0.
%
clear out
for i=1:length(max_cap_index_1)
           clear t
           t=find(max_cap_index_2==max_cap_index_1(i));
           if t
               out = max_cap_index_1(i);
               break
           else
               out=0;
           end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function maxi = max_cap_index(A, s)

% max_cap_index   Returns the indexes of the maximum entries in a vector.
%

clear AA a1 index maxi    %clearing the variables.
AA=(fix(A*100000))/100000;%Fixing to 5 decimal digits
[a1 index]=sort(AA);      %Ordering the capacities in ascendant order and obtaining the
indexes.
maxi = find(AA==a1(s));   %Finding the indexes of all orderings attaining maximum capacity.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [n c] = cumulative_elements(Cap_l);

% cumulative_elements    Returns the number of bins and the cumulative sum.
%

clear n n_elements c
n=[0:(max(Cap_l)/50):max(Cap_l)];%number of bins
n_elements=hist(Cap_l,n);        %obtaining the histogram of the Optimum ordering capacity.
c=cumsum(n_elements);            %obtaining the cumulative sum
```

IPRA: SNR gain of ordering (2 Tx antennas). Principal function.

```
function Gain_analysis
%
%Gain_analysis: Returns 3 graphs:
%               1- Cap. Vs. SNR for both orderings.
%               2- # of active streams Vs. SNR for both orderings
%               3- SNR gain of ordering (analytical and numerical) Vs. SNR
%

fprintf('\nPlease wait, Matlab is running the simulation ...\n')

%Initializing variables.

n=2;                %# of Rx antennas
m=2;                %# of Tx antennas
Pt=m;               %total power constraint
SNR_dB_start=-20;   %Start value of the SNR range (in dB)
SNR_dB_end=30;      %End value of the SNR range (in dB)
ind=0;
%H = 1/sqrt(2)*(randn(n,m) + 1i*randn(n,m))  %Random channel matrix, with entries iid CN-
(0,1).

H = [0.3335  -2.3299
     0.3914   -1.4491]

s=factorial (m);         %# of possible orderings of the channel matrix
G=zeros(n,m);            %Matrix used for the different orderings of the channel.
A=H;                     %Matrix used for manipulations.
SNR_dB_vector = [SNR_dB_start:1:SNR_dB_end] %vector with all the SNR values in dB.
SNR_vector = 10.^(SNR_dB_vector/10)         %vector with all the SNR values.
WF=zeros(1,length(SNR_dB_vector));          %vector to store the capacity of each ordering
under WF.
Strm_activ=zeros(1,length(SNR_dB_vector));  %vector to store the # of active streams of a
specific ordering under WF.
General = zeros(s,length(SNR_dB_vector));   %Matrix to store the capacity of each ordering
for each SNR value.
Active_Streams = zeros(s,length(SNR_dB_vector));%Matrix to store the # of active stream of
each ordering for each SNR value
P=[1:m];                 %Enumerating the columns of the channel matrix.
B=perms(P);              %All possible orderings of the columns of the channel matrix.

%squared norm of each of the columns of the channel matrix

for i=1:2
unp_pow(i)=norm(H(:,i))^2
end

%Generating the 1st ordering (column with highest norm last)

[a ind_asc]=sort(unp_pow) %Organizing the norms in an increasing order
H1=H(:,ind_asc)           %Channel matrix with the column with highest norm last.

%Generating the 2nd ordering (column with highest norm first)

ind1 = m:-1:1
ind_desc = ind_asc(ind1)  %Indexes of the columns of the ch matrix organized in a decreasing
way(according to the norm of his columns)
H2=H(:,ind_desc)          %Channel matrix with the column with highest norm first.

%Channel gains for both orderings

Pow1 = V_BLAST_Channel_Gains( m, H1)
Pow2 = V_BLAST_Channel_Gains( m, H2)


%Evaluating the capacity and the # of active streams of all orderings for
%all SNR values.

for i=1:s
```

```
    b=B(i,:);                 %Taking the j-th possible ordering.
    G=A(:,b);                 %Creating the channel matrix with the j-th ordering

    %obtaining the capacity and the # of active streams of the i-th ordering for all SNR
values

    for SNR_dB=SNR_dB_start:1:SNR_dB_end;
        ind = ind + 1;
        snr=10^(SNR_dB/10);
        [WF(ind) Strm_activ(ind)]=water_filling(G, m, snr,Pt);
    end
    General(i,(1:ind))= WF;                 %Saving the capacity values.
    Active_Streams(i,(1:ind))= Strm_activ; %Saving the # of active streams values
    ind=0                                   %Reset counter
    clear G b
end

%Allocating the data corresponding to the optimum ordering as the first row of each matrix.

if General(1,1)< General(2,1)
    General = General([2 1],:)
    Active_Streams = Active_Streams([2 1],:)
end

%Separating the data of the optimum and the suboptimum ordering

Cap_Opt = General(1,:)
Cap_Sub_Opt = General(2,:)
Active_Streams_Opt = Active_Streams(1,:)
Active_Streams_Sub_Opt = Active_Streams(2,:)

%Obtaining the analytical boundaries of the SNR regimes

[Thresh_1_dB Thresh_2_dB]=regimes(Pow1, Pow2);
Thresh_1 = 10^(Thresh_1_dB/10);

%Obtaining the analytical and the numerical gains for each SNR value.

for i=1:length(SNR_dB_vector)
    [Analytic_Gain(i) Numeric_Gain(i)] = Gains(SNR_vector(i), Cap_Opt(i),
Active_Streams_Opt(i), Active_Streams_Sub_Opt(i), Pow1,Pow2,Thresh_1);
end

%Plotting the Cap. vs. SNR for both orderings.

figure (1),
plot(SNR_dB_vector,General(1,(1:length(SNR_dB_vector))))
hold on
plot(SNR_dB_vector,General(2,(1:length(SNR_dB_vector))),'r')
plot
(0*General(1,(1:length(SNR_dB_vector)))+Thresh_1_dB,General(1,(1:length(SNR_dB_vector))))
plot
(0*General(1,(1:length(SNR_dB_vector)))+Thresh_2_dB,General(1,(1:length(SNR_dB_vector))))
legend('Optimum ord.', 'Suboptimum ord.')

%Plotting the # of active streams vs. SNR for both orderings

figure (2),
plot(SNR_dB_vector,Active_Streams_Opt)
hold on
plot(SNR_dB_vector,Active_Streams_Sub_Opt,'x')
plot (0*Active_Streams_Opt+Thresh_1_dB,General(1,(1:length(SNR_dB_vector))))
plot (0*Active_Streams_Opt+Thresh_2_dB,General(1,(1:length(SNR_dB_vector))))
legend('Optimum ord.', 'Suboptimum ord.')

%Plotting the SNR gain of ordering (analytical and numerical) vs. SNR

figure (3),
plot(SNR_dB_vector,Analytic_Gain)
hold on
plot(SNR_dB_vector,Numeric_Gain,'x')
```

```
plot (0*Analytic_Gain+Thresh_1_dB,General(1,(1:length(SNR_dB_vector))))
plot (0*Analytic_Gain+Thresh_2_dB,General(1,(1:length(SNR_dB_vector))))
legend('Analytic solution', 'Numerical solution')
end
```

IPRA: SNR gain of ordering (2 Tx antennas). Auxiliary functions.

```
function Pow = V_BLAST_Channel_Gains( m, G) %This function was used before (see IPRA:
%                                            Auxiliary functions).

% CONVENTIONALV_BLAST   Returns the channel gains of a system using
%                       conventional (without ordering) V-BLAST.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Cap Active] = water_filling(G, m, snr,Pt)

%water_filling:  Returns the capacity of the system using WF and the # of active streams.
%

clear Pow a
Pow = V_BLAST_Channel_Gains( m, G);  %Channel gain
a=sort(Pow);                         %Sorting the channel gains in an increasing order

%Water level for m active streams.

for j=m:-1:1
clear mio1 mio2 mio miu_m p p_g_snr Cap Active

mio1 = Pt/j;
mio2 = 1./a;
mio = (1/(j*snr))*sum(mio2);
miu_m = mio1+mio;           %Water level for m active streams.

%waterfilling algorithm

if miu_m >1/(snr*a(1))
    mio2=mio2*(1/snr);
    p=miu_m - mio2;           %vector storing the powers associated to each stream.
    p_g_snr = 1 + (p.*a*snr);
    Cap = log(p_g_snr);       %vector storing the capacity of each stream
    Cap=sum(Cap);             %Total capacity
    if length(a)==m
        Active=2
    else
        Active=1
    end
    break
else
    a(1)=[];                 %turning off the minimum stream;
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Thresh_1 Thresh_2]= regimes(Pow1,Pow2)
%
%regimes:  Returns the SNR values corresponding to the boundaries of the SNR regimes.

%High SNR Threshold: for all SNR above this value both orderings will operate with 2 active
%streams

one_over_Pow1 = 1./Pow1;
Thresh_2 = (1/2)*(one_over_Pow1(1)-one_over_Pow1(2))
Thresh_2 = 10*log10(Thresh_2)

%Low SNR Threshold: for all SNR below or equal to this value both orderings will operate with
1 %active stream.
```

```
one_over_Pow2 = 1./Pow2;
Thresh_1 = (1/2)*abs(one_over_Pow2(1)-one_over_Pow2(2));
Thresh_1 = 10*log10(Thresh_1)

%For all SNR values between the thresholds, the suboptimum ordering operates with 2 active
streams and the optimum with 1.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Analytic_Gain_dB Numeric_Gain_dB] = Gains(SNR, Cap_Opt, Active_Streams_Opt,
Active_Streams_Sub_Opt, Pow1, Pow2, Thresh_1)
%
%Gains   Returns the analytic and the numerical values of the SNR gain offered by the
%        ordering procedure.
%
clear Analytic_Gain_dB Numeric_Gain_dB Gain Numeric_Gain Analytic_Gain
Analytic_Gain_numerator Analytic_Gain_denumerator

if Active_Streams_Opt==1 && Active_Streams_Sub_Opt==1

    %numerical gain in the low SNR regime

    f=@(SNR_new)Cap_Opt - log(1 + 2*Pow2(1)*SNR_new)   %Equation to find the numerical gain
in the low SNR regime
    SNR_new=fzero(f,[SNR 10000])                       %Solving the equation to find the
numerical gain

    if SNR_new <= Thresh_1
        Numeric_Gain = abs(SNR_new/SNR);
        Numeric_Gain_dB = 10*log10(Numeric_Gain);      %Numerical gain in dB

        %Analytic gain in the low SNR regime

        Gain=Pow1(2)./Pow2;                            %Analytical SNR gain at low SNR regime
        Analytic_Gain_dB=abs(10*log10(min(Gain)));     %Analytical SNR gain in dB

    %if SNR_new > Thresh_1 the gain has to be calculated as in the case of
    %the intermediate SNR regime. However, for this SNR value both orderings
    %are operating with 1 active stream.

    else

        %Equation to find the numerical gain in the intermediate SNR regime

        f=@(SNR_new)Cap_Opt - log(((2*Pow2(1)*Pow2(2)*SNR_new + Pow2(1) +
Pow2(2))/(2*Pow2(2)))*((2*Pow2(1)*Pow2(2)*SNR_new + Pow2(1) + Pow2(2))/(2*Pow2(1))));
        SNR_new=fzero(f,[SNR 10000]);%Solving the equation to find the numerical gain
        Numeric_Gain = abs(SNR_new/SNR);
        Numeric_Gain_dB = 10*log10(Numeric_Gain);%Numerical gain in dB

        %Equation to find the analytic gain

        Analytic_Gain_numerator = ((2*2*Pow1(1)*Pow1(2)*(1+(2*Pow1(2)*SNR))).^(1/2))-
(Pow2(1)+Pow2(2));         %Numerator of the analytic SNR gain equation at the intermediate
SNR
        Analytic_Gain_denumerator=(2*Pow1(1)*Pow1(2)*SNR);%Divisor of the SNR gain equation
        Analytic_Gain=Analytic_Gain_numerator/Analytic_Gain_denumerator;%SNR gain at
intermediate SNR regime.
        Analytic_Gain_dB=10*log10(Analytic_Gain); %Analytic SNR gain in dB

    end
else
    if Active_Streams_Opt==1 && Active_Streams_Sub_Opt==2

        %Analytic gain in the intermeduiate SNR regime

        Analytic_Gain_numerator= ((2*2*Pow1(1)*Pow1(2)*(1+(2*Pow1(2)*SNR))).^(1/2))-
(Pow2(1)+Pow2(2));%%Numerator of the SNR gain equation at intermediate SNR
        Analytic_Gain_denumerator=(2*Pow1(1)*Pow1(2)*SNR);%Divisor of the SNR gain equation
        Analytic_Gain=Analytic_Gain_numerator/Analytic_Gain_denumerator;%SNR gain at
intermediate SNR regime.
```

```
        Analytic_Gain_dB=10*log10(Analytic_Gain); %SNR gain in dB

        %numerical gain in the intermediate SNR regime

        f=@(SNR_new)Cap_Opt - log(((2*Pow2(1)*Pow2(2)*SNR_new + Pow2(1) +
Pow2(2))/(2*Pow2(2)))*((2*Pow2(1)*Pow2(2)*SNR_new + Pow2(1) + Pow2(2))/(2*Pow2(1))))
        SNR_new=fzero(f,[SNR 10000]);
        Numeric_Gain = abs(SNR_new/SNR);
        Numeric_Gain_dB = 10*log10(Numeric_Gain);
    else

        %Analytic gain in the high SNR regime

        Analytic_Gain_numerator=(2*Pow1(1)*Pow1(2)*SNR)+Pow1(1)+Pow1(2)-Pow2(1)-Pow2(2);%
Numerator of the SNR gain equation
        Analytic_Gain_denumerator=(2*Pow1(1)*Pow1(2)*SNR);%Divisor of the SNR gain equation
        Analytic_Gain=Analytic_Gain_numerator/Analytic_Gain_denumerator;%SNR gain at high SNR
        Analytic_Gain_dB=abs(10*log10(Analytic_Gain)); %SNR gain in dB

        %Numerical gain in the high SNR regime

        f=@(SNR_new)Cap_Opt - log(((2*Pow2(1)*Pow2(2)*SNR_new + Pow2(1) +
Pow2(2))/(2*Pow2(2)))*((2*Pow2(1)*Pow2(2)*SNR_new + Pow2(1) + Pow2(2))/(2*Pow2(1))));
        SNR_new=fzero(f,[SNR 10000]);
        Numeric_Gain = abs(SNR_new/SNR);
        Numeric_Gain_dB = 10*log10(Numeric_Gain);
    end
end
```

IPA: Principal function.

```
function principal_IPA
%
%principal_IPA:   Returns the CDF graphs of the Foschini ordering, the optimum ordering and
the   %                unordered detection under the IPA. It also verifies if the Foschini
ordering   %                 offers the maximum capacity.
%
%Initializing data

SNR_dB=10;                                        %received SNR (dB).
m=4;                                              %number of transmit antennas.
n=4;                                              %number of receive antennas.
M=100;                                            %number of channel realizations.
snr=10^(SNR_dB/10);                               %received SNR(absolute value).
Cap_Unord_IPA_l=zeros(1,M);                       %vector to store the capacity of the
unordered detection for each channel realization under the IPA.
Opt_Cap_IPA_l=zeros(1,M);                         %vector to store the capacity of the
optimum ordering for each channel realization under the IPA.
Cap_H_foschini_IPA_l=zeros(1,M);                  %vector to store the capacity of the
Foschini ordering for each channel realization under the IPA.
tic;
fprintf('\nPlease wait, Matlab is running the simulation ...\n')

%generating M channel realizations and saving the capacity offered by each ordering strategy.

for l=1:M
    clear Cap_Unord_IPA Opt_Cap_IPA Cap_H_foschini_IPA H;
    H = 1/sqrt(2)*(randn(n,m) + 1i*randn(n,m));%Random channel matrix, with entries iid CN-
(0,1).

    %Obtaining the capacity of each ordering for the given channel realization.

    [Cap_Unord_IPA Opt_Cap_IPA Cap_H_foschini_IPA ] =bruta_IPA(n, m, H, snr);

    %Saving the capacity of each ordering for the given channel realization.

    Cap_Unord_IPA_l(l)=Cap_Unord_IPA;
    Opt_Cap_IPA_l(l)=Opt_Cap_IPA;
```

```
    Cap_H_foschini_IPA_l(l)=Cap_H_foschini_IPA;
end

%Verifying if the Foschini ordering achieve the maximum capacity for all channel
realizations.

Difference = Opt_Cap_IPA_l-Cap_H_foschini_IPA_l;
nonzero=find(Difference);

%Obtaining the number of bins and the cumulative sum for the capacity chieved by each
ordering of %interest.

[n_Cap_Unord_IPA c_elements_Cap_Unord_IPA] = cumulative_elements(Cap_Unord_IPA_l);
[n_Opt_Cap_IPA c_elements_Opt_Cap_IPA] = cumulative_elements(Opt_Cap_IPA_l);
[n_Cap_H_foschini_IPA c_elements_Cap_H_foschini_IPA] =
cumulative_elements(Cap_H_foschini_IPA_l);

elapsed_time=toc;
elapsed_time=elapsed_time/60

%Plotting the CDF graphs

figure(1),
semilogy(n_Cap_Unord_IPA,(c_elements_Cap_Unord_IPA/M),'-
',n_Opt_Cap_IPA,(c_elements_Opt_Cap_IPA/M),'-
k',n_Cap_H_foschini_IPA,(c_elements_Cap_H_foschini_IPA/M),'p');
legend('Unordered IPA','Opt IPA','Foschini IPA')
title(sprintf('Empirical Capacity CDFs P[C<=R]  SNR = %d dB   Channel realizations %d   %dx%d
Time: %g min',10*log10(snr),M,m,n,elapsed_time));
ylabel('Prob');
xlabel('R')
grid on
```

IPA: Auxiliary functions.

```
function [Cap_Unord_IPA Opt_Cap_IPA Cap_H_Foschini_IPA] =bruta_IPA(n, m, H, snr)
%
%BRUTA_IPA   Returns the capacity of the unordered detection, the capacity of the Foschini
%            ordering and the optimum capacity under the IPA.
%

clear  P s O  A G B b IPA Cap_Unord_IPA a Opt_Cap_IPA H_Foschini Cap_H_Foschini_IPA ind
Opt_CH_Ord;
s=factorial (m);               %# of all possible orderings of the channel matrix.
IPA=zeros(1,s);                %row vector to store the capacity of every ordering under the
IPA.
G=zeros(n,m);                  %matrix used for the different orderings of the channel
matrix.
A=H;
P=[1:m];                       %Enumerating the columns of the channel matrix.

%Evaluating the capacity for all orderings (for a given channel realization) under the IPA.

B=perms(P);                    %All possible orderings of the columns of the channel matrix.
for i=1:s
    b=B(i,:);                  %Taking the i-th ordering.
    G=A(:,b);                  %Creating the channel matrix with the i-th ordering.
    IPA(i)= Capacity_under_IPA( m, G, snr);  %Obtaining the cap of the i-th ordering.
    clear G b
end

%Capacity of the unordered detection under the IPA

Cap_Unord_IPA = IPA(1);

%Capacity of the optimum ordering under the IPA

[a ind]=sort(IPA);             %Ordering the capacities in ascendant order.
Opt_Cap_IPA=a(s);              %Obtaining the optimum capacity.
```

```
%Capacity of the Foschini ordering under the IPA

H_Foschini=orderingS(n,m,H);                                %Obtaining Foschini ordering.
Cap_H_Foschini_IPA=Capacity_under_IPA( m, H_Foschini, snr); %Capacity of the Foschini
ordering.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cap_IPA = Capacity_under_IPA( m, G, snr)

% Cap_Fractional_WF   Returns the capacity under the IPA for a given channel realization.
%
%
clear  Cap_IPA Pow g ;            %clearing the variables.

%Obtaining the after-processing channel gains.

Pow = V_BLAST_Channel_Gains( m, G);

%Channel inversion

g = harmonic_mean_perstream_gain(m, Pow);  %Obtaining the harmonic mean per stream power
gain.
Cap_IPA = m*log(1 + snr*g);                %Obtaining the capacity.
Cap_IPA = (fix(Cap_IPA*100000))/100000;   %Fixing the capacity value to 5 decimal digits.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Pow = V_BLAST_Channel_Gains( m, G) %This function was used before (see IPRA:
%                                            Auxiliary functions).

% V_BLAST_Channel_Gains   Finds the channel gains of a system using conventional (without
%                         ordering) V-BLAST.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [n c] = cumulative_elements(Cap_l); %This function was used before (see IPRA:
%                                             Auxiliary functions).

% cumulative_elements     Returns the number of bins and the cumulative sum.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

function g = harmonic_mean_perstream_gain( m, a)

%harmonic_mean_perstream_gain   Returns the harmonic mean per-stream power gain (inverse of %
%                               the sum of the inverse of each of the channel gains)

clear g           %clearing the variables.
g = 1./a;         %inversion of each of the channel gains
g = 1/m*sum(g);   %sum of the inversion of each of the channel gains
g = 1/g;          %harmonic mean perstream power gain

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Hod = orderingS(n,m,H)

% ordering.m   Returns the Foschini ordering.
%              Columns of H are sorted to maximize after-processing SNR at each step
%
clear Z Aw Out Hod W;             % clearing projection variables
Hod = zeros(n,m);                 % ordered matrix
Z = H;                            % channel matrix to be used in the detection
for i = 1:m-1                     % Detection Process, i-th step
    clear Aw As ind X Q P;
    wj = zeros(n,m);
    for j = 1:m-i+1               % Orthogonal Projection Procedure
        X = Z;                    % channel matrix to be reduced by 1 column
        X(:,j) = [];              % deleting i-th column of H
```

```
        Q = null(X');             % orthonormal basic of the null space of X'
        P = Q*Q';                 % orthogonal projection matrix
        wj(:,j) = P*Z(:,j);       % orthogonal projection of i-th column of H
        Aw(j) = norm(wj(:,j))^2;  % length of the projected vector
    end
    [As,ind] = sort(Aw);          % sorting the elements of Ah in ascending order
    Hod(:,i) = Z(:,ind(m-i+1));   % optimum column is inserted into the ordered matrix
    Z(:,ind(m-i+1))= [];          % detected Tx column is deleted
end

Hod(:,m) = Z;                     % the last column of the ordered matrix
```

URA: Principal function.

```
function principal_URA
%
%principal_UPA:   Returns the CDF graphs of the Foschini ordering, the optimum ordering and %
%                 the unordered detection under the URA. It also verifies if the Foschini % %
%                 ordering offers the maximum capacity.
%

SNR_dB=-20;                         %received SNR (dB).
m=3;                                %number of transmit antennas.
n=3;                                %number of receive antennas.
M=100;                              %number of channel realizations.
snr=10^(SNR_dB/10);                 %received SNR(absolute value).
Opt_Cap_URA_l=zeros(1,M);           %vector to store the capacity of the optimum ordering for
each channel realization under the IPA.
Cap_H_foschini_URA_l=zeros(1,M);    %vector to store the capacity of the Foschini ordering
for each channel realization under the URA.
Unord_Cap_URA_l=zeros(1,M);         %vector to store the capacity of the unordered detection
for each channel realization under the URA.
tic;
fprintf('\nPlease wait, Matlab is running the simulation ...\n')

%generating M channel realizations and saving the capacity offered by each
%ordering strategy

for l=1:M
    clear Unord_Cap_URA Opt_Cap_URA Cap_H_Foschini_URA H;
    H = 1/sqrt(2)*(randn(n,m) + 1i*randn(n,m));%Random channel matrix, with entries iid CN-
(0,1).

    %Obtaining the capacity of each ordering for the given channel realization

    [Opt_Cap_URA Cap_H_Foschini_URA Unord_Cap_URA] =bruta_URA(n, m, H, snr);

    %Saving the capacity of each ordering for the given channel realization

    Opt_Cap_URA_l(l)=Opt_Cap_URA;
    Cap_H_foschini_URA_l(l)=Cap_H_Foschini_URA;
    Unord_Cap_URA_l(l)=Unord_Cap_URA;
end

%Verifying if the Foschini ordering achieve the same capacity as the optimum ordering for all
channel realizations.

Difference = Opt_Cap_URA_l-Cap_H_foschini_URA_l
nonzero=find(Difference)

%Obtaining the number of bins and the cumulative sum for the capacity achieved by each
ordering of interest.

[n_Opt_Cap_URA c_elements_Opt_Cap_URA] = cumulative_elements(Opt_Cap_URA_l);
[n_Cap_H_foschini_URA c_elements_Cap_H_foschini_URA] =
cumulative_elements(Cap_H_foschini_URA_l);
[n_Unord_Cap_URA c_elements_Unord_Cap_URA] = cumulative_elements(Unord_Cap_URA_l);
```

```
elapsed_time=toc;
elapsed_time=elapsed_time/60

%Plotting the CDF graphs

figure(1),
semilogy(n_Opt_Cap_URA,(c_elements_Opt_Cap_URA/M),'-
r',n_Cap_H_foschini_URA,(c_elements_Cap_H_foschini_URA/M),'kd',n_Unord_Cap_URA,(c_elements_Un
ord_Cap_URA/M));
legend('Opt URA','Foschini URA', 'Unordered URA')
title(sprintf('Empirical Capacity CDFs P[C<=R]  SNR = %d dB   Channel realizations %d   %dx%d
Time: %g min',10*log10(snr),M,m,n,elapsed_time));
ylabel('Prob');
xlabel('R')
grid on
```

URA: Auxiliary functions.

```
function [Opt_Cap_URA Cap_H_Foschini_URA Unord_Cap_URA] =bruta_URA(n, m, H, snr)
%
%
%BRUTA_URA   Returns the capacity of the unordered detection, the capacity of the Foschini
%            ordering and the optimum capacity under the URA. It also verifies if the
%            Foschini ordering is equal to the optimum ordering.
%
clear  P s O  A G B b a Opt_Cap_URA H_Foschini Cap_H_Foschini_URA;
s=factorial (m);              %# of all possible orderings of the channel matrix
O=zeros(1,s);                 %row vector to store the capacity of every ordering under the
URA.
G=zeros(n,m);                 %matrix used for the different orderings of the channel.
A=H;
P=[1:m];                      %Enumerating the columns of the channel matrix.

%Evaluating the capacity for all orderings (for a given channel realization) under the URA.

B=perms(P);                   %all possible orderings of the columns of the channel matrix.
for i=1:s
    b=B(i,:);                 %taking the i-th ordering.
    G=A(:,b);                 %creating the channel matrix with the i-th ordering
    O(i)= Capacity_under_URA( m, G, snr);     %obtaining the capacity of the i-th ordering.
    clear G b
end

%Capacity of the unordered detection under the URA

Unord_Cap_URA=O(1);           %capacity of the unordered channel using URA+UPA.

%Capacity of the optimum ordering under the IPA

[a ind]=sort(O);              %ordering the capacities using URA+UPA in ascendent order.
Opt_Cap_URA=a(s);             %obtaining the optimum capacity

%Creating the optimum ordering

b=B(ind(s),:);                %taking the index of the optimum ordering.
Opt_CH_Ord=A(:,b);            %optimum ordering


H_Foschini=orderingS(n,m,H);   %obtaining Foschini ordering
Cap_H_Foschini_URA=Capacity_under_URA( m, H_Foschini, snr); %capacity of the Foschini
ordering under URA.

%Verifying if the Foschini ordering is optimum.

if Opt_Cap_URA~=Cap_H_Foschini_URA
   H_Foschini
   Opt_CH_Ord
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cap_URA_UPA = Capacity_under_URA( m, G, snr)

% Cap_Fractional_WF   Returns the capacity under the URA for a given channel realization.
%

clear  Cap_URA_UPA C Pow;                        %clearing the variables.

%Obtaining the after-processing channel gains.

Pow = V_BLAST_Channel_Gains( m, G);

%Obtaining the capacity

for j=1:m
C(j)=log(1 + snr*(Pow(j)));                      %capacity at each of the steps.
end
Cap_URA_UPA=m*min(C);                            %total capacity for URA+UPA.
Cap_URA_UPA=(fix(Cap_URA_UPA*100000))/100000;  %fixing the capacity value to 6 decimal
digits.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Pow = V_BLAST_Channel_Gains( m, G) %This function was used before (see IPRA:
%                                            Auxiliary functions).
%
% V_BLAST_Channel_Gains   Finds the channel gains of a system using conventional (without
%                         ordering) V-BLAST.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [n c] = cumulative_elements(Cap_l); %This function was used before (see IPRA:
%                                            Auxiliary functions). %
% cumulative_elements     Returns the number of bins and the cumulative sum.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Hod = orderingS(n,m,H)     %This function was used before (see IPA: Auxiliary
%                                    functions).
%
% ordering.m   Returns the Foschini ordering.
%              Columns of H are sorted to maximize after-processing SNR at each step
```