Università degli Studi di Padova
Facoltà di Ingegneria
Corso di Laurea in Ingengneria delle Telcomunicazioni

Analysis and System Design of a cooperative wireless network based on Interference Alignment

RELATORE
Prof. Michele Zorzi

CORRELATORE
Dr. Francesco Rossetto

LAUREANDO
Giulio Baldo
Matr. N. 602024

Anno Accademico 2010/2011

Knowledge consists in the search for truth.
But it is not the search for certainty.

KARL POPPER

Contents

Abstract

Interference Alignment is one of the newest interference management techniques and it tries to align interference signals at the receiver side to achieve in this way better performance in terms of throughput. At present, there are only very few results about this technique in cooperative networks. In literature all the best results are achieved under ideal conditions, like fast fading, completely uncorrelated channel, arbitrary long temporal slots, but we aim to analyze the scenario in normal and real conditions and we think that network cooperation can help to better approximate these ideal conditions.

We aim to verify if, thanks to cooperation, better results in terms of throughput and convergence speed are reachable.

At first we study and analyze various scenarios, extending the current models in both temporal and space domain, that is to let the system work in more than one temporal slot and with cooperative entities. Since results in closed solution are difficult to be achieved, we prefer to analyze these several cooperative setups thanks to simulations exploiting iterative algorithms. We try to generalize the empirical results, in an analytical way when possible.

From the results we obtain, we can not affirm that an effective capacity improvement is achievable, because in all the analyzed scenarios the correlation on channel, even exploiting cooperation, limits the performance. Anyway some results about capacity are obtained. If in cooperative scenarios further hypothesis of mutual channel knowledge of the sources are stated, configurations over this kind of setup can reach better properties and so raise their throughput. We can finally affirm that cooperation is useful to improve convergence speed of the Interference Alignment algorithms and this is explained not only thanks to simulation results, but also thanks to an

analytical interpretation.

We think that especially this final result could be exploited and be useful as an improvement on applications using Interference Alignment.

Sommario

L'Interference Alignment è una delle più nuove tecniche di gestione dell' interferenza ed ha l'obiettivo di allineare al ricevitore i segnali interferenti, ottenendo così migliori prestazioni in termini di throughput. Al momento ci sono solo pochi studi riguardo questa tecnica in reti cooperative. In letteratura tutti i migliori risultati sono ottenuti sotto condizioni ideali, come fast-fading, canali completamente incorrelati, slot temporali arbitrariamente lunghi. Noi intendiamo analizzare lo scenario in condizioni normali e reali, ritenendo che proprio la cooperazione permetta di approssimare meglio le condizioni ideali.

Il nostro intentento è di verificare se, tramite la cooperazione, sono ottenibili migliori risultati in termini di throughput e velocità di convergenza.

All'inizio del nostro lavoro, studiamo e analizzaiamo vari scenari, estendendo i modelli correnti sia da un punto di vista temporale che dal punto di vista spaziale, il che significa permettere al sistema di poter lavorare su più di uno slot, introducendo al contempo entità cooperative. Poichè è difficile ottenere risultati in forma chiusa, preferiamo analizzare diversi setup cooperativi tramite simulazioni impieganti algoritmi iterativi. Quando possibile, cercheremo di generalizzare i risultati raggiunti empiricamente in maniera analitica.

Dai risultati ottenuti, è difficile poter afferamare che un miglioramento capacitivo sia raggiungibile, perchè la correlazione di canale, anche in scenari cooperativi, limita irremidiabilmente le prestazioni. Tuttavia alcuni risultati rigurado la capacità sono comunque possibili. Nello specifico, se in scenari cooperativi sono poste ulteriori ipotesi di mutua conoscenza del canale da parte delle sorgenti, le configurazioni in queste condizioni ottengono migliori

proprietà, riuscendo ad incrementare in questa maniera il proprio throughput. Infine si può affermare che la cooperazione è utile per migliorare la velocità di convergenza degli algoritmi di Interference Alignment e ciò è dimostrato non solo empiricamente, tramite le simulazioni, ma anche attraverso una interpretazione analitica.

Riteniamo che specialmente quest'ultimo risultato possa essere sfruttato ed essere utile come miglioramento di applicazioni che utilizzano Interference Alignment.

Zusammenfassung

Interference Alignment ist eine der neuesten Interferenz-Management-Techniken, die versucht, Interferenzsignale im Empfänger abzugleichen, um so einen besseren Durchsatz zu erreichen. Momentan existieren von dieser Technik nur sehr wenige Ergebnisse im Bezug auf kooperative Netzwerke. In der Literatur werden die besten Ergebnisse unter idealen Bedingungen erreicht, wie schnellem Fading, vollständig unkorrelierten Kanälen und beliebig langen Zeitschlitzen. Wir aber versuchen, das Szenario unter normalen Bedingungen zu analysieren, und glauben, dass kooperative Netzwerke helfen können, diese idealen Bedingungen besser zu approximieren.

Wir wollen zeigen, dass mit Hilfe der Kooperation bessere Ergebnisse bezogen auf Durchsatz und Konvergenzgeschwindigkeit erreichbar sind.

Zuerst analysieren wir verschiedene Szenarien und erweitern die aktuellen Modelle sowohl im zeitlichen als auch im räumlichen Bereich, damit das System mit mehr als einem Zeitschlitz und mit kooperativen Einheiten arbeiten kann. Da Ergebnisse in geschlossenen Lösungen schwierig zu erreichen sind, ziehen wir es vor, diese kooperativen Setups mit Simulationen zu analysieren, die iterative Algorithmen verwenden. Wir versuchen, die empirischen Ergebnisse auf eine analytische Weise zu verallgemeinern, sofern dies möglich ist.

Anhand der Ergebnisse, die wir erhalten, können wir zwar nicht behaupten, dass eine effektive Kapazitätssteigerung möglich ist, da in allen betrachteten Szenarien die Korrelation in den Kanälen die Leistung begrenzt. Trotzdem werden einige Erkenntnisse zur Kapazität deutlich. Wenn in kooperativen Szenarien weitere Annahmen über gegenseitige Kanalinformationen gemacht werden, können Konfigurationen mit dieser Art von Einstellung bessere Eigenschaften erhalten und so ihren Durchsatz steigern. Wir können

letztendlich behaupten, dass Kooperation nützlich ist, um die Konvergenzgeschwindigkeit des Interference-Alignment-Algorithmus zu verbessern, und das wird nicht nur anhand der Simulationsergebnisse erklärt, sondern auch durch eine analytische Interpretation.

Wir glauben, dass vor allem dieses letzte Ergebnis genutzt werden kann und als Verbesserung für Anwendungen, die Interference Alignment verwenden, dienen kann.

Acknowledgment

At first I would like to thank Prof. Michele Zorzi from University of Padua, to have given me the possibility to perform my master thesis at DLR and to have supported me during these six months.

Obviously I can not forget all my colleagues at DLR especially my supervisor, Dr. Francesco Rossetto, who spent so much time with me: thanks to you I have understood what ”doing research” means. You also taught me a method to face the problems, which is not only valid for the specific topic analyzed in this thesis, but can be used as a forma mentis.

I have to mention also all my office-mates in the famous ”Student office” for the unforgettable time I passed there: Jawad, Tudor (the smallest one!), Firat (thanks to me you learnt a very good Italian), Stefan (thank you for the fassung!), Carlos (I will miss your music and your parties) and of course Davide, with who, I think, I shared (almost) everything in the last six months and he always put up with me. A big GRAZIE to Giuliano and Tomaso: with your reference letter, I think I will find a job everywhere! Finally I would like to thank all the guys at DLR and the students of other departments with who I had a lot of fun: Manu, Maria, Carlos, Hanno, Michael and all the others.

No puedo olvidar a todos mis amigos de Erasmus que he conocido en München: gracias a vosotros he hecho muchas fiestas, me he divertido muchísimo y al final he aprendido este ”español”. Un agradecimiento muy grande a Jorge, Claudia, Valerio, Teresa, Andrea, los dos Danilos, Gianni, Alex, Charlotte, Pier, Laura, Lucia, Sonia, Silvia, Angelo y a todos los demás (había mucha gente!).

Tuttavia i ringraziamenti più doverosi non possono che essere fatti in italiano. Sicuramente un pensiero va a tutti i miei compagni di corso, con i quali ci siamo aiutati e supportati a vicenda durante questi cinque anni: citarvi uno ad uno sarebbe impossibile, ma un bel grazie collettivo ve lo meritate anche voi!

Uno dei ringraziamenti più sentiti va ai miei amici, quelli veri, quelli di sempre, quelli che sento sempre volentieri, quelli con i quali vado in vacanza, quelli che mi son veramente mancati gli scorsi sei mesi, quelli che con una parola sono veramente felice di frequentare e fiero di poter dire che sono miei AMICI. Un grazie sentito a Francesco (è la prima e l'ultima volta che ti chiamo così), Pippo, Marzio, Live, Petti, Nicola, Maria, Ga, Cristina, Anna, Dega, Clizia, Barbara, Stefano, Vg e tutti gli altri con cui mi son sempre divertito.

Infine, ma sicuramente non per importanza, il ringraziamento maggiore va alla mia famiglia ed in particolare ai miei genitori che mi han sempre supportato economicamente e moralmente, che non mi hanno mai fatto mancare il loro appoggio, anche nei momenti più difficili: è grazie soprattutto a voi se posso ormai affermare di essere giunto al termine del mio percorso di studi.

Baone, Padova
April 12th, 2011

GIULIO BALDO

Chapter 1

Introduction and problem statement

Interference Alignment (IA) is one of the newest interference management (IM) techniques. Traditionally interference has been regarded as a damaging element in the design of telecommunications systems and it has been avoided as much as possible, often by means of orthogonal access schemes like TDMA or FDMA. Some research trends have lately tried to exploit its structure to limit its extent. One remarkable example is multiuser detection, but there a new approach (called IA) has been very recently proposed, which could in principle increase the capacity of wireless networks to a far greater extent than most other IM approaches but at present there are only very few results about this technique in cooperative networks. This thesis aims to explore this topic, analyzing several aspects of the theme in both a simulative and analytical way.

The work was performed at the *Deutsches Zentrum für Luft- und Raumfahrt* (*DLR*) in Oberpfaffenhofen, Germany.

In this chapter a brief introduction about MIMO, Interference Alignment and cooperation is present. Finally the a subsection about the problem statement describing what we aim to do in this theses concludes the chapter.

1.1 MIMO overview

To improve robustness or data rate over wireless communications, there exist many techniques like OFDM or CDMA. On the following lines a short introduction over Multiple Input Multiple Output (MIMO) system is done, which constitutes the basis of the thesis.

In conventional radio systems, transmitter and receiver have just one antenna each, this is called Single Input Single Output (SISO) system.

A MIMO system employs a number of transmit antennas M_t and receive antenna M_r usually greater than one. These systems are said to achieve *diversity gain*, *power gain* and *space-multiplexing gain* [1]. We will briefly introduce the first two ones, while we will focus more on the last one which is the most important and it is mostly exploited in our work.

MIMO systems are said to achieve *diversity gain* when the data is coded and transmitted through different antennas, in such a way to increase the power in the channel. The techniques that do an efficient encoding are called *space-time coding* techniques and one of the most famous is called Alamouti code [1]. It is a space-time encoding scheme at first designed for two transmit antennas, but an extension to several antennas is possible. The encoding matrix is the following:

$$X = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix} \tag{1.1}$$

where s_i are the transmitting symbols.

In this way at the receiver side the two received signals are orthogonal, which leads to a simpler decoding. The result of these techniques is an improvement of Signal Noise Ratio (SNR) and increase on the reliability of wireless link.

Moreover receive antennas can also provide *power gain*. It is also called array gain and refers to the beamforming capability of a multiple antenna array. The main idea is that it is possible to direct radiated energy toward the receiver in a steered beam, improving channel performance and increasing the throughput. One of the most important application [1] is on Multiple Input Single Output (MISO) systems on on which power gain is provided thanks to transmit beamforming. In this case $M_r = 1$ while $M_t > 1$ and the channel H can be configured as a vector of dimensions $1 \times M_t$. The main idea is to send all the transmit power in the direction of the channel vector *H,* information sent in the orthoganl direction will be nulled out anyway. In this way calling the transmitting vector signal x we would like:

$$x = \frac{H}{||H||}\tilde{x} \tag{1.2}$$

where $\tilde{x}$ is the so called pre-image vector and is the signals vector before the mapping on the antennas (see System model). In this way the received signal is proportional to the norm of H and a power gain is obtained. Notice that to do so, the trasmitter has to know the channel, that means that exploits Channel-State-Information (CSI).

We introduce now *space-multiplexing gain.* Each transmit antenna sends its own independent signal (say t_1, t_2, $...t_{M_t}$) and all of them will be transmitted simultaneously in the same frequency band. As a consequence, sharing the same channel, every antenna receives not only the direct component from the respective antenna, but also the indirect components intended for the others. By calling the received signal at each antenna as r_j, the system could be represented in the following way, while a scheme is present in Fig. 1.1

$$\begin{aligned}
r_1 &= h_{1,1}t_1 + h_{1,2}t_2 + \ldots + h_{1,M_t}t_{M_t} \\
r_2 &= h_{2,1}x_1 + h_{2,2}x_2 + \ldots + h_{2,M_t}t_{M_t} \\
&\vdots \\
r_{M_r} &= h_{1,1}x_1 + h_{1,2}x_2 + \ldots + h_{1,M_t}t_{M_t}
\end{aligned}$$

As it can be seen from the above set of equations, in making their way from the transmitter to the receiver, the independent set of signals $\{t_1, t_2, ...t_{M_t}\}$ are all combined and traditionally this "combination" has been treated as interference. Otherwise, by looking at the system in an another way, the channel can be represented as a matrix of dimension $M_r \times M_t$:

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,M_t} \\ h_{2,1} & h_{2,1} & \dots & h_{2,M_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M_r,1} & h_{M_r,1} & \dots & h_{M_r,M_t} \end{bmatrix} \tag{1.3}$$

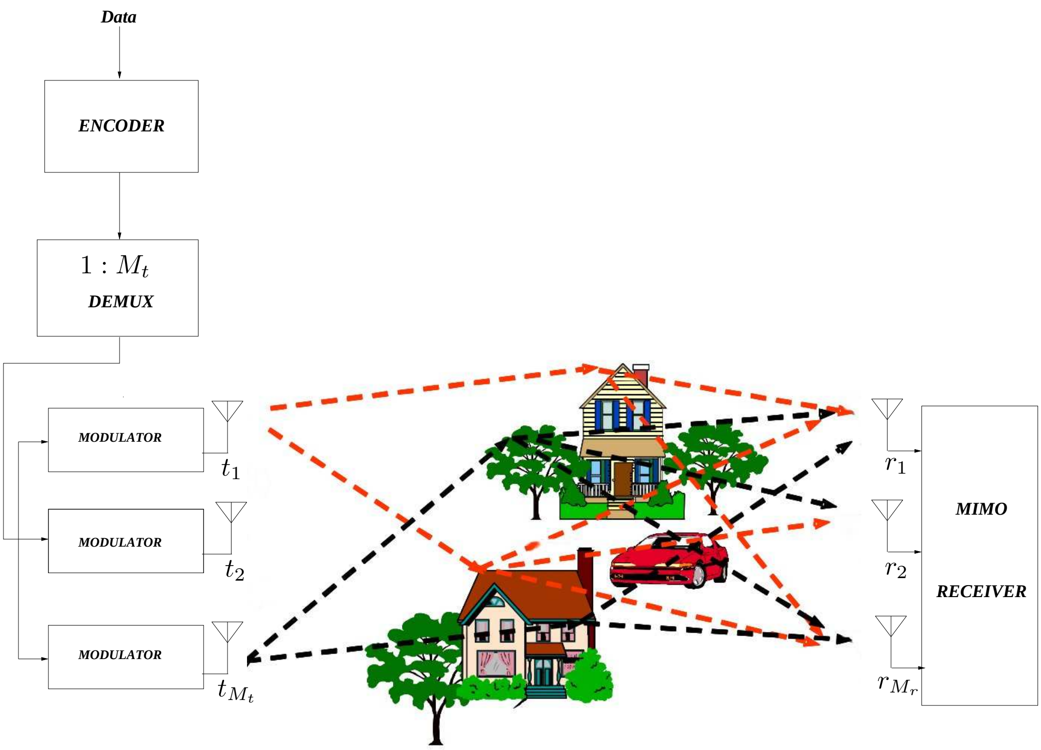

Figure 1.1: Generic MIMO system

In this way to recover the transmitted data t_i from the received ones r_j, it is necessary to estimate the individual channel weights $h_{i,j}$ and to reconstruct H. Having estimated H, the multiplication of the vector y=$[r_1\, r_2 \dots r_{M_r}]$ with the pseudo-inverse of H produces vector $x = [t_1\, t_2 \dots t_{M_t}]$:

$$x = H^{\dagger} y \tag{1.4}$$

This is equivalent to solving a set of M_r equations in M_t unknowns.

To be resolvable, the system has to contain more variables than equations and so $M_t \geqq M_r$, which means that the system must have the same or more transmit than receive antennas. In this way the throughput increases linearly with every pair of antennas added to the system without an increase of the bandwidth.

The basic idea of MIMO is to exploit the so called "*spatial diversity*" to increase the throughput: to ensure H is invertible with high probability, a MIMO system requires an environment rich on multipath, so that the matrix could be as uncorrelated as possible. As a consequence MIMO is an example of a system where line of sight invalidates performances while fading is necessary.

Another way to obtain "spatial diversity" [1] consists to use the angular domain of the antennas used. Adaptive antenna arrays intensify spatial multiplexing using narrow beams. Defining the angular resolution as the minimum angle thanks to whom two signals that arrive with this angle can be resolved; the angular resolution of a linear antenna array is dictated by its length: an array of length L provides a resolution of *1/L*. Transmit and receiver antenna arrays of length L_t and L_r partition the angular domain into $2L_t \times 2L_r$ bins of unresolvable multipaths as showed in Fig. 1.2 also picked from [1].

In real application the types of antennas preferred are called "Smart Antennas" and are usually divided into two groups as in Fig. 1.3:

- Phased Array Systems (Switched Beamforming) with a number of finite number of fixed predefined patterns

- Adaptive Array Systems (Adaptive Beamforming) with an indefinite number of patterns adjusted to the scenario in real time

The second typology is often preferred because the beamformer is adjusted in real-time as the moving of the terminal, while in the first case the beams are fixed. Obviously the cost and the complexity in second case are higher than the first one.

MIMO analysis and performances are different depending on the hypothesis made on channel. Usually it is assumed that the receiver knows the

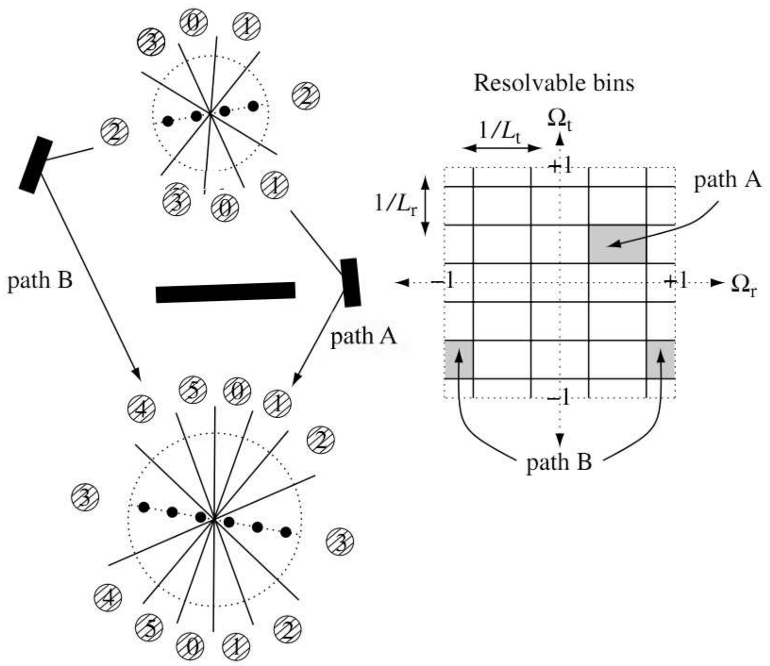

Figure 1.2: The MIMO channel in the angula domain

channel, while the the transmitter is devoid of such information. MIMO systems where both the transmitter and receiver know the channel are called MIMO with Channel state Information (CSI) at both communication ends. This hypothesis could be really strong, but it is often used depending on the environment: if the channel is slowly time varying, for example like the satellite channel the hypothesis can be realistic. In the case of MIMO with CSI more knowledge is introduced on the system and better performance (e.g. capacity) are achieved.

1.1.1 System model

Delving more into the argument, a MIMO systems can be seen as in Fig. 1.4 and described in [1]. The transmitter has a variable number of streams or information flows, shortly described by the vector $\tilde{x}$ of dimension $S \times 1$. The number of streams could be greater than one but not more than the

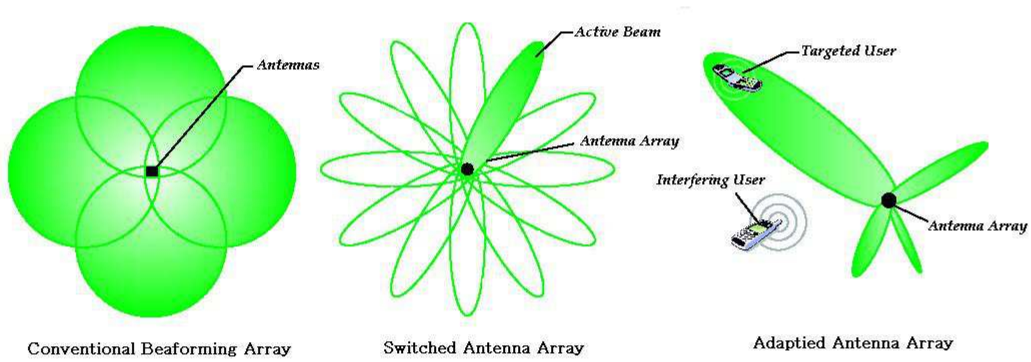

Figure 1.3: Examples of swiched and adaptive beamforming

minimum between the number of transmit and receive antennas:

$$1 \leqq S \leqq \min\{M_r,\, M_t\} \tag{1.5}$$

Each stream is a sequence of modulated symbols and they are mapped, thanks to the *precoding matrix* V ($dim(V)$ $=M_t \times S$), in the transmitted signal, the vector x of dimensions $M_t \times 1$. This is the signal sent from the antennas of the transmitter through the wireless channel. The channel between the transmitter and the receiver could be modeled by the matrix H of dimensions $M_r \times M_t$ equal to the one represented in (1.3). The receiver captures the received signal y of dimensions $M_r \times 1$ which is the signal from the related transmitter corrupted by the thermal noise. At the receiver side, streams are reconstructed through the decoding matrix U ($dim(U)$ $=S \times M_r$), and the vector $\widetilde{y}$ is obtained with dimensions $S \times 1$. Each symbol at the receiver side passes through a normal decoder to detect the original transmitted one.

Notice that no assumption on the channel distribution is done. The matrix H presented before is totally generic, but usually the channel will be characterized with suitable distributions like Rayleigh, Loo, etc. In this work, we use a Rayleigh fading, even if the topology is motivated by satellite scenarios.

It is possible to describe the system thanks to the following formulas:

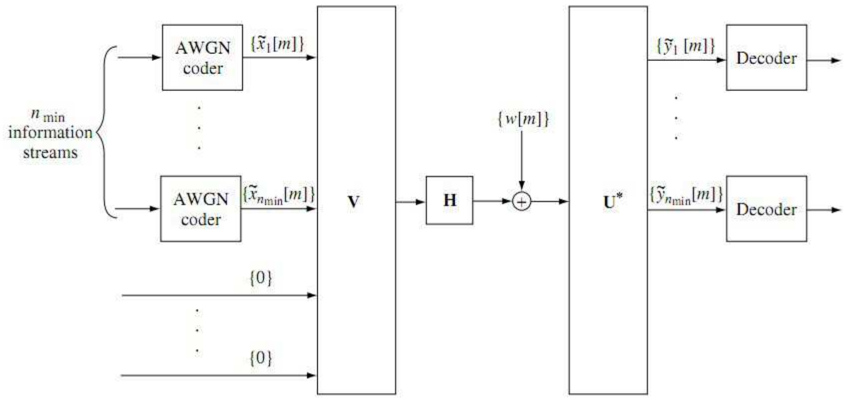

Figure 1.4: Architecture for MIMO communication

$$x = V\,\tilde{x} \tag{1.6}$$

$$y = H\,x + w \tag{1.7}$$

$$\tilde{y} = U^H y \tag{1.8}$$

Passing to analyze capacity, it is useful to remark first a fundamental theorem made by Shannon and Hartley stating that the capacity of a SISO system is

$$C = B\log_2(1 + SNR) \tag{1.9}$$

where B is the bandwidth and SNR is the signal to noise ratio

It is demonstrated that the ergodic capacity of a $M_r \times M_t$ i.i.d. fading MIMO channel H with receiver CSI is

$$C(SNR) = \mathbb{E}\left[\log\det\left(I_{M_r} + \frac{SNR}{M_t}\,HH^H\right)\right] \tag{1.10}$$

At high SNR the capacity is approximately equal to $\min\{M_{t,} M_r\} \times \log(SNR)$ bit/s/Hz, while at low SNR Eq. (1.10) could be approximated by $M_r \log(SNR)$ bit/s/Hz. Finally if $M_r = M_t = M$ capacity could be approximately expressed by $Mc^*(SNR)$ at both high and low SNR, where

$$c^*(SNR) = 2\log(1 + SNR - \frac{1}{4}F(SNR)) - \frac{\log e}{4SNR}F(SNR)$$

and

$$F(SNR) = \left(\sqrt{4SNR+1} - 1\right)^2$$

When $M_r = M_t = M$ it is possible to state that the capacity increases linearly with the M for the entire SNR range.

All the demonstrations are present in [1].

Usually more than one MIMO transmitter and MIMO receiver are present. The starting point of this thesis is a special network called K *network* as in [2] which will be analyzed in next chapters. The network topology is such that there are K transmitter-receiver couples where the *i-th* transmitter wants to communicate only with the *i-th* receiver and both of them have the characteristics explained before. In this case the total system can be seen as:

$$x_l = V_l \widetilde{x}_l \tag{1.11}$$

$$y_k = \sum_{k=1}^{K} H_{k,l} x_l + w_k \tag{1.12}$$

$$\widetilde{y}_k = U_k^H y_k \tag{1.13}$$

where l is the subscript of transmitters, k the one for receivers and all the other symbols have been already introduced before and on here are only pointed out for the particular transmitter/receiver. The big change is with Eq. (1.12): the received signal is corrupted not only by noise but also by interfering signals from other transmitters. These systems are often called Multi-User MIMO, and, except for the multiple access and the broadcast channel, a formula for the capacity region is difficult to be obtained. Any way approximations are present achievable thanks to the concept of *degrees of freedom* that will be introduced in next section.

1.2 Introduction to Interference Alignment

The main concept introduced and analysed in this work is the concept of Interference Alignment (IA) signal processing technique.

When a receiver has to decode its useful signal, one or more interference signals could be present. The basic idea of IA is to try to align an interference signal on another one as they could be "confused" and seen at the receiver side just as a single one. The consequence of IA is to achieve in this way a higher throughput. In the further sections the concept of IA will be analyzed in more detail and it will be explained how cooperation and IA can operate together.

1.2.1 IA: basic information theory considerations

In order to fully appreciate the benefits of IA, it is first necessary to introduce the concept of *Degree of Freedom* (DoF). The DoF approach provides an approximation of the channel capacity which is valid at asymptotically high signal-to-noise ratio (SNR). A network has d degrees of freedom if and only if the sum capacity of the network can be expressed as:

$$C = d \log(SNR) + o(\log(SNR)) \tag{1.14}$$

Each interference-free signaling dimension yields $\log(SNR)+o(\log(SNR))$ and so the degrees of freedom in a network could be seen as the number of resolvable signal space dimensions.

The task of IA, from an information theory point of view, is to maximize the dimension of the useful signal, thanks to the alignment of interference. Cadambe and Jafar introduce in [2], the so-called K user interference channel. It is made of of K transmitters and K receivers each of them can be equipped with both a single or a multiple antenna and all the channel coefficients picked i.i.d. from a continuous distribution. In this network each transmitter wants to communicate with its own receiver. In the case of a single antenna nodes and without any loss of generality, the assumption is that each transmitter 1, 2,..., K has an independent message W_1, W_2, ..., W_K intended for receiver 1, 2, ... K. Respectively assuming that total power per transmitters is equal to ρ and indicating the size of each message as $|W_i(\rho)|$, the authors say that the rates $\mathfrak{R}_i(\rho) = \frac{\log|W_i(\rho)|}{t_0}$ are achievable by choosing an appropriately large t_0. In this way the capacity region $C(\rho)$ of the K user interference channel is said to be the set of all *achievable* rate tuples $\mathbf{R}(\rho) = (\mathfrak{R}_1(\rho), \mathfrak{R}_2(\rho)..., \mathfrak{R}_\mathrm{K}(\rho))$. The authors demonstrate that the achievable DoF for the global system is $K/2$.

The converse of the previous statement follows the following lemma which provides an outer bound on the degrees of freedom:

$$\max_D d_i + d_j \le \limsup_{\rho\to\infty} \sup_{\mathbf{R}(\rho)\in C(\rho)} \frac{R_i + R_j}{\log(\rho)} \le 1,\ \forall\, i,\ j \in \{1,\ 2\ldots K\}\,,\ i \neq j \tag{1.15}$$

where D is the degree of freedom region. The previous lemma can be extended in the following way:

$$\max_D \sum_{i,\,j\in\{1,\,2\ldots K\},\,i\neq j} (d_i + d_j) \le \sum_{i,\,j\in\{1,\,2\ldots K\},\,i\neq j} 1 \tag{1.16}$$

$$\Rightarrow \max_D d_1 + d_2 + \ldots + d_K \le K/2 \tag{1.17}$$

which demonstrate the statement. The demonstration of all lemmas are present in [2].

For the same network topology and in the same conditions, but with MIMO nodes with $M_r = M_t = M$ antennas, the authors demonstrate that the achievable DoFs are $KM/2$.

In [3], the same authors introduce the so-called X-network (all transmitters have at least a packet for all receivers). This network has a number of transmitters and receivers which is in general different, i.e. $K_r \neq K_t$, where K_r is the number of receivers K_t is the number of transmitters. They demonstrate that in presence of completely uncorrelated channel coefficients, an upper bound on the DoF of the system exists and it is $MK_rK_t/(K_r + K_t - 1)$, where M is the number of antennas for each node. This result is achieved by the demonstration of the following theorem:

$$\begin{aligned} D^{out} \; = \; & \{[(d_{ji})] : \forall (l,\, k) \in \{1,\, 2,\, ...,\, K_t\} \times \{1,\, 2,\, ...,\, K_r\}\,, \\ & \sum_{q=1}^{K_r} d_{ql} + \sum_{p=1}^{K_t} d_{kp} - d_{kl} \leq \max(M_r,\, M_t)\} \end{aligned} \tag{1.18}$$

The theorem states that the number of degrees of freedom achieved by all messages associated with transmitter l and receiver k called D^{out}, is upper bounded by $\max(M_r,\, M_t)$. To have an upper bound of the total number of degrees of freedom of the X channel with K_t transmitters, K_r receivers and one antenna each node, the authors use the following corollary:

$$\max_{[(d_{ji})] \in D} \sum_{k \in \{1,2,...,K_r\},\, l \in \{1,2,...,K_t\}} d_{kl} \leq \frac{K_r K_t}{(K_r + K_t - 1)} \tag{1.19}$$

The demonstration of (1.19) is directly obtained by (1.18) by summing all the $K_r K_t$ inequalities and setting $M_r = M_t = 1$ for all transmitters and receivers. Demonstration of theorem (1.18) is present in the same paper [3]. A second corollary is also present in [3] and it explicits the total number of degrees of freedom of the X network, D_Σ, when all the nodes have M antennas. It is derived from previous equation and states the following:

$$\frac{M K_r K_t}{(K_r + K_t - 1/M)} \leq D_\Sigma \leq \frac{M K_r K_t}{(K_r + K_t - 1)} \tag{1.20}$$

Notice that all these results for the X network and K network are "only" upper bounds. The authors also show in the papers that in *asymptotic conditions*, that is to say when temporal slots are arbitrary long and fast-fading is present, there are interference alignment schemes which are able to approach within any $\varepsilon > 0$ these upper bounds.

1.2.2 IA: a geometric interpretation

As written before, IA is a signal processing technique that seeks to align interference at the receiver side. In this way more interfering signals could be treated just as a single one. In order to better understand this concept, IA can be analyzed also from a geometric point of view. In this case it is necessary to interpret the signals as elements of a suitable vector space. For instance each received signal could be seen as a vector on $\mathbb{C}^{M_r}$. Simply if more interference signals are aligned together, they count as only one because they occupy just one dimension in the signal space. To better clarify the previous affirmations, an example from [2], is proposed in the following lines. Just consider a K network with $K = 3$, $M_t = M_r = 1$, $S_{1,} = 2$, $S_2 = 1$, $S_3 = 1$ and fast-fading conditions,as in Fig. 1.5. As explained before the system could achieve at most $3/2$ degrees of freedom. This is an asymptotic result, in fact the number of degrees of freedom of this example follows the law:

$$DoF = \frac{3n+1}{2n+1} \tag{1.21}$$

where the numerator is the total number of packet symbol decoded by all the receivers, the denominator is the total number of symbol extension or temporal slot and n is an integer number. When $n \to \infty$, $DoF \to 3/2$, which confirms what written before. Authors say that it is like to have a useful space and a "waste basket space for interference. When n is a finite number, it is not possible to reach $1/2$ degree of freedom per user. It is like to say that we have an "overflow" space where interference signal and useful signal are present. Fortunately this space dimensions become smaller as n increases, that is to say that the upper bound is approximated within an $\varepsilon > 0$ which becomes smaller as n increases or otherwise as the number of slot increases.

Here the system is studied after three temporal slot ($n = 1$) and so the receiving space dimension is three. Let us define $V_l^{(j)}$ as the precoding vector of the j-th stream of the l-th transmitter (note that $V_l^{(j)}$ is the j-th column vector of matrix V_l). If the stream index is not specified it means that the transmitter has only one stream. Let us choose V_2 randomly. The first receiver wants to decode $V_1^{(1)}$ and $V_1^{(2)}$ and its useful space dimension will be equal to two, with only 1 dimension free for interference. It has to align the interference from transmitters 2 and 3:

$$H_{1,2}V_2 = H_{1,3}V_3 \Rightarrow V_3 = (H_{1,3})^{-1}H_{1,2}V_2$$

At receiver 2 there is only 1 useful signal, and the interference space dimension is 2 with 3 interference signals and therefore it has to align at least two of them:

$$H_{2,3}V_3 = H_{2,1}V_1 \Rightarrow V_1^{(1)} = (H_{2,1})^{-1}H_{2,3}(H_{1,3})^{-1}H_{1,2}V_2$$

Similar to 2, receiver 3 has the same constraints:

$$H_{3,2}V_2 = H_{3,1}V_1^{(2)} \Rightarrow V_1^{(2)} = (H_{3,1})^{-1}H_{3,2}V_1^{(2)}$$

Note that each receiver has a number of interference free dimensions equal to the number of streams that each one has to decode, thus the SIR related to each receiver is infinite and IA is successful. In this way the decoding of 4 useful signals with a 1 antenna system and 3 slots is possible while this data rate would not be achievable in usual systems without IA. In this case the system is said to have a total of $4/3$ degrees of freedoms.

The basic idea is to set precoding and decoding matrices such that at each receiver the projection of all interference on the vectors U_k is null, while the projection of the useful signal is not.

Other similar examples are covered in other papers like [3] and [4]. In [3] an example of *X-network* with single antenna nodes is reported. In this scenario all the transmitters have a packet to all the receivers. As written before, in this case it is demonstrated that a total amount of $K_rK_t/(K_r + K_t - 1)$ degrees of freedom for the system is reachable and the alignment is done over K_t*+1* symbols. This means that for example, if $K_t = 2$ and $K_r = 2$, IA can operate on this network and it is possible to decode 4 signals in the total system, 2 signals each receiver over an extension of 3 symbols. In this case both the receiver have two information signals and two interference signals each while dimension of signal space is 3 due to the number of slots and the number of receiving antennas.

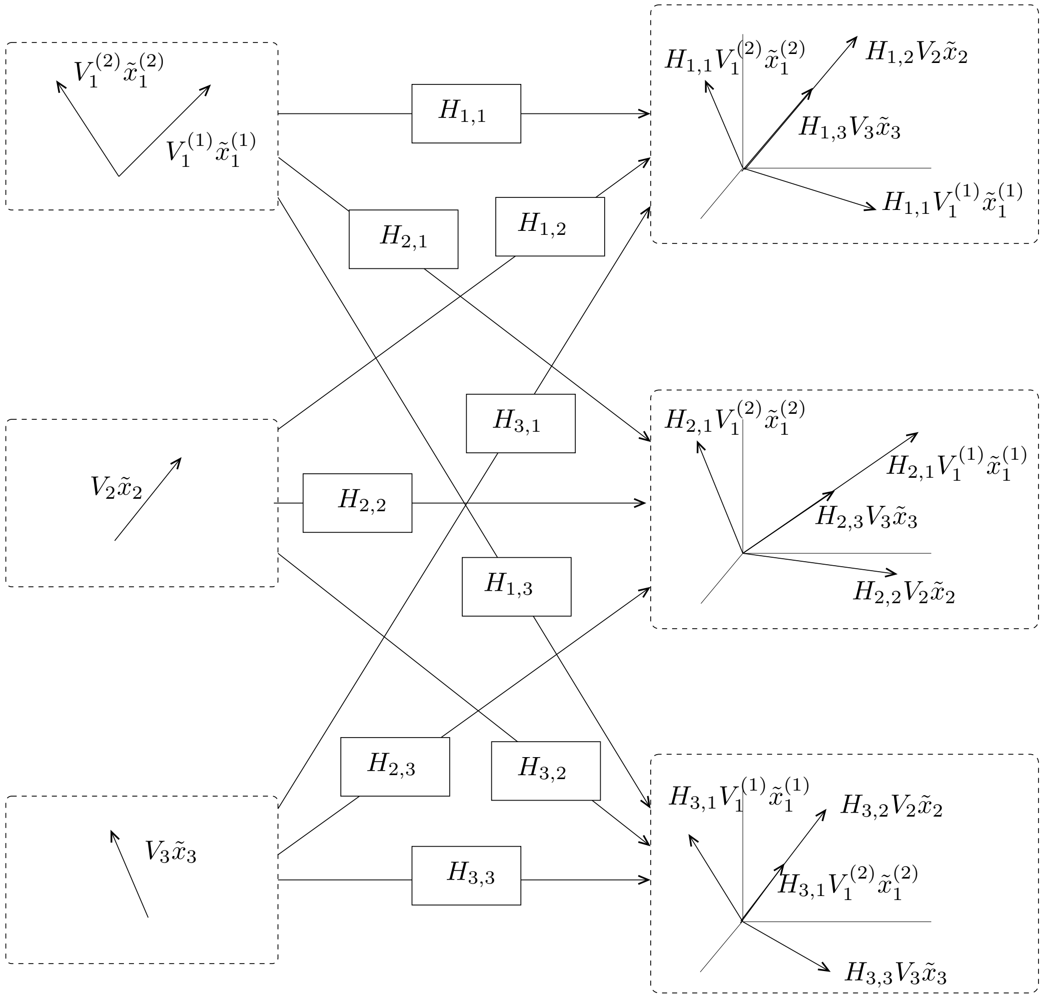

Figure 1.5: Interference alignment on the three-user interference channel to achieve 4/3 degrees of freedom

The scheme is summarized in by Fig. 1.6.

Let us focus first on the first receiver. It has to decode packets associated to precoders $V_1^{(1)}$ and $V_2^{(1)}$. That means that the dimension for useful signal must be equal to two. As a consequence interference signal dimension must be equal to one and so two interference signal $V_1^{(2)}$ and $V_2^{(2)}$, must be aligned together as the decode could be successful. As in previous example, let us pick $V_1^{(2)}$ randomly. The alignment is successful if:

$$H_{1,2}V_2^{(2)} = H_{1,1}V_1^{(2)} \Rightarrow V_2^{(2)} = H_{1,2}^{(-1)}H_{1,1}V_1^{(2)} \tag{1.22}$$

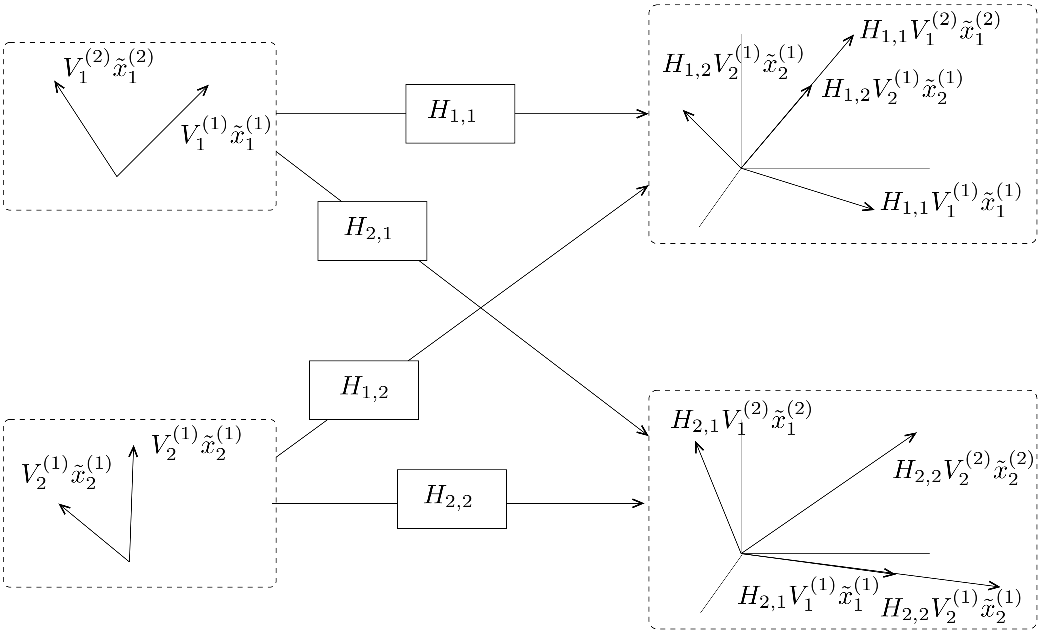

Figure 1.6: X-network with $K_t = 2$ and $K_r = 2$

At the second receiver it is exactly the same: the alignment of the two interference signal is necessary. This operation is the described by the following expression:

$$H_{2,2}V_2^{(1)} = H_{2,1}V_1^{(1)} \Rightarrow V_2^{(1)} = H_{1,2}^{(-1)}H_{2,1}V_1^{(1)} \tag{1.23}$$

In this way a total throughput of four symbol packets over a three symbol extension is reached.

Another paper on which IA is treated in a simple way with a great quantity of examples is [4].The main aspect that differs from the approaches analyzed before deals with the possibility of receivers to communicate with each other. In their work the authors imagine that receivers are MIMO access point (AP) and they are linked each other through an Ethernet connection. As a consequence only the first transmitters use interference alignment, they decode some of the packets, communicate them to the neighbors which cancel these decoded ones and decode the others so that at the end all the receivers can know all the packets. The algorithm presented here is called by the authors “Interference Alignment and Cancellation” (IAC). In the assumptions of the authors and differently from the examples presented before, the system works in just one temporal slot but as written before the AP exploit MIMO and as a consequence, the signal space is equal to the number of receiving antennas. Another difference from previous works is that all receivers want to know all packets as there are some packets to be received by only one receiver and not the others.

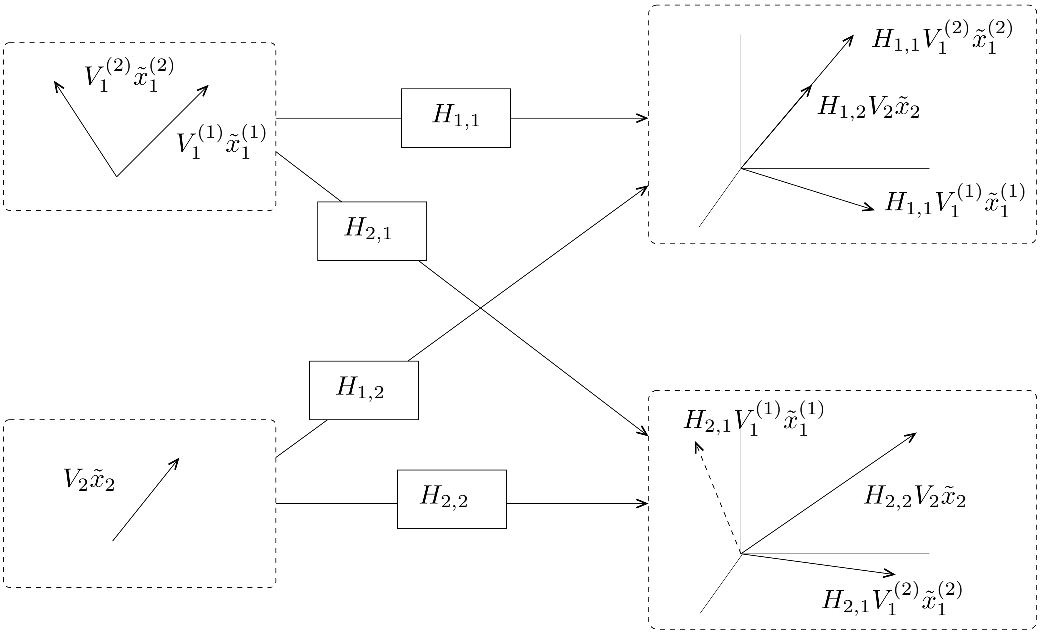

Figure 1.7: IAC with two receivers and two antennas equipment

On the following lines we comment the example of Fig. 1.7 reported in [4]. There are two receivers equipped with 2 antennas and so each one has a signal space equal to two. The first receiver decides to decode $\tilde{x}_1^{(1)}$ and so has to align vectors associated to $\tilde{x}_1^{(2)}$ and $\tilde{x}_2$. To do so, picked a random vector for $V_1^{(2)}$:

$$H_{1,2}V_2 = H_{1,1}V_1^{(2)} \Rightarrow V_2 = H_{1,2}^{(-1)}H_{1,1}V_1^{(2)} \tag{1.24}$$

In this way the decoding of $\tilde{x}_1^{(1)}$at the first receiver is successful. This packet is passed through the Ethernet cable to the second receiver which can cancel it, hence it has only two packets to decode and, being the receiving space dimension exactly equal to two, the decoding is successful. These new decode packets will be sent to the first receiver and in this way all the receiver will know all the packets. Note that differently from [3, 2], the receivers have to wait the decoding of the others to make cancellation possible. As stressed by the authors, both IA and Cancellation are necessary to make the system to work. In the example presented the throughput achievable could be even more. On the network presented in Fig. 1.8 there are three receivers and a total amount of four packets. The first AP decides to decode packet $\tilde{x}_1^{(1)}$ and so aligns vectors associated to $\tilde{x}_1^{(2)}$, $\tilde{x}_2$ and $\tilde{x}_3$. After decoding $\tilde{x}_1^{(1)}$ is passed to the second receiver which cancel it and decodes $\tilde{x}_1^{(2)}$ thanks the alignment of vectors associated to $\tilde{x}_2$ and $\tilde{x}_3$. $\tilde{x}_1^{(1)}$ and $\tilde{x}_1^{(2)}$ are passed to the last receiver, which cancel them and performs the decoding of the last packets.

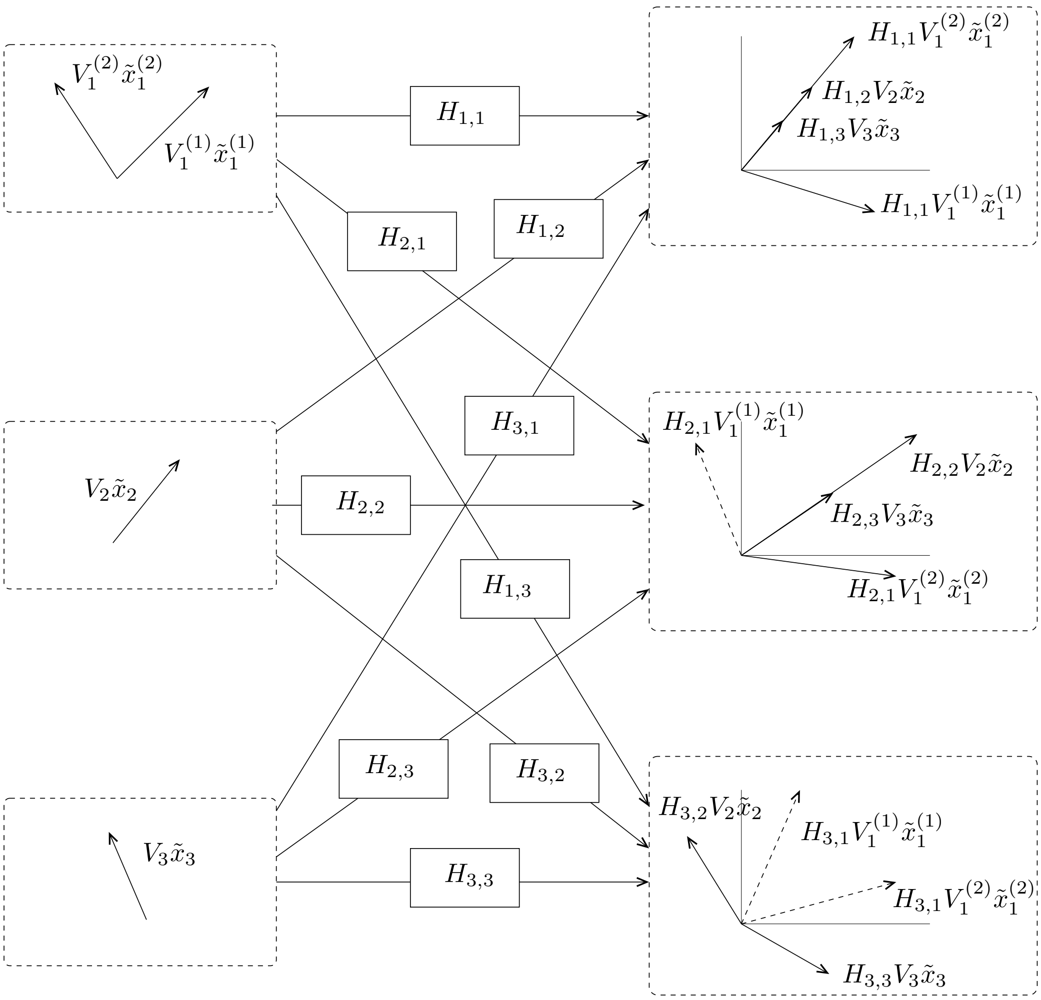

Figure 1.8: IAC with three receivers and two antennas equipment

The system in Fig. 1.8 may be explained through the following formulas:

$$H_{1,1}V_1^{(2)} = H_{1,2}V_2 = H_{1,3}V_3 \tag{1.25}$$

$$H_{2,2}V_2 = H_{2,3}V_3 \tag{1.26}$$

On here the system is fully constrained because the system of equations expressed in the formula below is a system of two equations in two variables:

$$\begin{cases} H_{1,2}V_2 = H_{1,3}V_3 \\ H_{2,2}V_2 = H_{2,3}V_3 \end{cases} \tag{1.27}$$

From (1.27) V_2 and V_3 are uni-vocally determined and so:

$$V_1^{(2)} = H_{1,1}^{(-1)}H_{1,2}V_2 = H_{1,1}^{(-1)}H_{1,3}V_3 \tag{1.28}$$

As written before the system is fully constrained and all the DoFs are saturated such that with a two antenna equipment, no more packets can be decoded. If the system has more than two antennas a the system behaves in a similar way but the throughput, obviously, grows. In this paper the author demonstrate, simply by induction starting from $M = 2$ (the example in Fig. 1.8 and reported before), that the throughput achievable is equal to $2M$, where M is the number of antennas of all the equipment.

In all the illustrated examples IA alignment was possible. This is not always true: if the system is too much constrained, such that there are too many packets in the network or too many interfering transmitters etc., IA is not reachable. Another important hypothesis on the system configuration is that all the channel matrices are known by all the receivers and transmitters or by other words, they have acquired Channel State Information (CSI). This could be a strong hypothesis, but it is often used: for example previous examples make this assumption and may be realistic if the channel is slowly time varying, for example like the satellite channel.

1.2.3 Examples of implementations of IA

How to realize and implement IA? Usually solutions in a closed form are very difficult to find and numerical ones are necessary. In [5] a similar solution is presented. It is an iterative algorithm that at each iteration optimizes precoding and decoding vectors. This approach is also called *Zero-Forcing* and aims to minimize the function:

$$\sum_{k=1}^{K}\sum_{l\neq k}^{K}||H_{k,l}V_l - C_k C_k^H H_{k,l}V_l||^2 \tag{1.29}$$

where the matrix C_k is the so-called matrix of interference for the receiver *k,* which has dimension $M_r - S_k \times M_r$, and it generates the orthogonal complement to the subspace spanned by U_k. The metrics to be minimized could be seen as the projection of the interference onto the subspace of the useful signal (i.e. it is the residual interference) and so it could also be rewritten as:

$$\sum_{k=1}^{K}\sum_{l\neq k}^{K}||U_k U_k^H H_{k,l}V_l||^2 \tag{1.30}$$

At each step the matrices C_k are computed by choosing the $M_r - S$ dominant eigenvectors of the following matrices:

$$\sum_{l\neq k}^{K} H_{k,l}V_l V_l^H H_{k,l}^H \tag{1.31}$$

while matrices V_l are computed by choosing the S least dominant eigenvectors of :

$$\sum_{k\neq l}^{K} H_{k,l}^H (I_{M_r} - C_k C_k^H) H_{k,l} \tag{1.32}$$

These results are derived from two lemmas explained and demonstrated in [5]. The first establishes that given K arbitrary matrices $A_k \in \mathbb{C}^{N\times q}$, the p-dimensional subspace $\mathfrak{U}$, with minimum overall Euclidean distance to the columns of all the A_k, has orthonormal basis U where the columns of U

are the eigenvectors associated with the p largest eigenvalues of $\sum A_k A_k^H$. The second one states that given K arbitrary p-dimensional subspaces $\mathfrak{U}_\mathfrak{k}$ with respective orthonormal bases U_k and $M \times N$ matrix B, the matrix V such that $A = BV$, $V \in C^{N \times q}$, that minimizes the squared Euclidean distance from the columns of A to the subspaces has columns equal to the eigenvectors corresponding to the q minimum eigenvalues of $\sum_k B(I - U_k U_k)B$.

In the following lines an other example of iterative algorithm presented in [6] is reported. It is similar to the one presented in [5] but in the optimization a new term is introduced. After a random initialization of precoder matrices V_l, at each step the interference matrices C_k can be:

$$C_k = \arg\min_{C_k} \sum_{l=1,\, l\neq k}^{K} ||H_{k,l}V_l - C_k C_k^H H_{k,l} V_l||^2 + w\,||H_{k,k}V_k - U_k U_k^H H_{k,k} V_k||^2 \tag{1.33}$$

where like before the first term is the residual interference in the signal space and the other is the residual useful signal in the interference space (as before $U_k = C_k^\perp$). w is an empirical positive calculated weight and not further specified by the authors. The novelty with respect to the previous algorithm is obviously the second term which tries to lead to a C_k that yields a better SINR than the previous algorithm computed. Recalling that $U_k U_k^H = C_k (C_k^\perp)^H = I - C_k C_k^\perp$, Eq. (1.33) can be rewritten as

$$C_k = \arg\min_{C_k} \sum_{l=1,\, l\neq k}^{K} ||H_{k,l}V_l - C_k C_k^H H_{k,l} V_l||^2 + w\,||C_k C_k^H H_{k,k} V_k||^2 \tag{1.34}$$

and then using the trace operator and its properties:

$$C_k = \arg\max_{C_k} \operatorname{tr}\left\{ C_k^H \left[\sum_{l=1,\, l\neq k}^{K} H_{k,l} V_l H_{k,l}^H V_l^H - w H_{k,k} V_k H_{k,k}^H V_k^H \right] C_k \right\} \tag{1.35}$$

For the same reasons of the first algorithm, the solution to the above optimization problem corresponds to choosing the columns of C_k to be the

$(N_k - S_k)$ dominant eigenvalues of:

$$\sum_{l=1,\, l\neq k}^{K} H_{k,l} V_l V_l^H H_{k,l}^H - w H_{k,k} V_k V_k^H H_{k,k}^H \tag{1.36}$$

Precoders optimization is absolutely analogous to decoders one and the result corresponds to choosing the columns of C_k to be the S_k least dominant eigenvalues of:

$$\sum_{l=1,\, l\neq k}^{K} H_{k,l}^H (I - C_l C_l^H) H_{k,l} + w H_{k,k}^H C_k C_k^H H_{k,k} \tag{1.37}$$

The authors prove also that their algorithm converges just saying that the overall optimization is the sum of two objective functions and so each step an improvement is done and so convergence is granted.

The last algorithm presented in here is picked by [7] and constitutes something new because for the first time a *Max-SINR* algorithm is presented, differently from the two above that were different implementations of *Zero-Forcing*. First the authors introduce the definition of SINR for the k^{th} receiver and its j^{th} stream:

$$SINR_k^{(j)} = \frac{(U_k^{(j)})^H H_{k,k} V_k^{(j)} (V_k^{(j)})^H H_{k,k}^H U_k^{(j)}}{(U_k^{(j)})^H B_k^{(j)} U_k^{(j)}} \tag{1.38}$$

where

$$B_k^{(j)} = \sum_{l=1}^{K} \sum_{j=1} H_{k,l} V_l^{(j)} (V_l^{(j)})^H H_{k,l}^H - H_{k,k} V_k^{(j)} (V_k^{(j)})^H H_{k,k}^H + N_0 I \tag{1.39}$$

is the interference plus noise covariance matrix. The authors state that the vector $U_k^{(j)}$ that maximizes $SINR_k^{(j)}$ is given by:

$$U_k^{(j)} = \frac{(B_k^{(j)})^{-1} H_{k,k} V_k^{(j)}}{||(B_k^{(j)})^{-1} H_{k,k} V_k^{(j)}||} \tag{1.40}$$

Notice that in the paper it is not really specified the derivation of the previous formula. Describing the algorithm, the authors specify the following steps:

- random fix the precoders V_k
- begin iteration
- compute $B_k^{(j)}$ as (1.39) for stream j of receiver k $\forall k \in \{1, 2, ..., K\}$, $j \in \{1, 2, ..., S_k\}$
- calculate $U_k^{(j)}$ as (1.40) for stream j of receiver k $\forall k \in \{1, 2, ..., K\}$, $j \in \{1, 2, ..., S_k\}$
- reverse communication direction and set precoding vectors equal to decoding ones: $V_k = U_k \; \forall k \in \{1, 2, ..., K\}$
- recompute $B_k^{(j)}$ as (1.39) for stream j of receiver k $\forall k \in \{1, 2, ..., K\}$, $j \in \{1, 2, ..., S_k\}$
- recalculate $U_k^{(j)}$ as (1.40) for stream j of receiver k $\forall k \in \{1, 2, ..., K\}$, $j \in \{1, 2, ..., S_k\}$
- reverse communication direction and set precoding vectors equal to decoding ones: $V_k = U_k \; \forall k \in \{1, 2, ..., K\}$
- repeat until convergence

Notice that this algorithm requires channel reciprocity, which must be verified in the network of interest.

1.3 Cooperative networks

The main idea of cooperative wireless networks is that wireless agents, called users, try to increase their quality of service in terms of bit error rates, block error rates or outage probability thanks to cooperation. In a cooperative communication system, there is a certain set of users (called relays or cooperators) that may retransmit the traffic of other terminals [9]. In satellite context relays are also called *Complementary Ground Component* (CGC) or *Ancillary Ground Component* (AGC). In some cases they may also send traffic of their own. At first glance it might seem that such approach causes a loss of data rate in the system, however the spectral efficiency of each user improves because, thanks to cooperation diversity, the channel code rate can be increased. What is important on cooperative communication is that for each user there exist at least one partner providing different data paths. In this way a trade off is observed, but several studies give evidence that cooperation is useful especially in high SNR channels. Following the survey [10], in the next subsections the main cooperation techniques are reviewed and a simple scheme is present in Fig. 1.9 picked from [10].

1.3.1 Amplify-and-forward

In this method each user receives a noisy version of the signal from its partner. The user than amplifies the received signal, affected by the noise, and retransmit it to the base station. The receiver compares the two signals and it can make a better decision on the detection of information. In amplify-and-forward method is assumed that the base station knows the interuser channel coefficients to do an optimal decoding. This method is for sure the simplest one and the computational capability required is the lowest of all the typologies of cooperation, but it is not very efficient.

1.3.2 Decode-and-forward

In this method each user receives the analog signal from the partner, decodes it and then retransmits the detected bits. To clarify this method an example from [11] is reported. A simple code-division-multiple access is imagined (CDMA) and two users are present, cooperating each others. We denote with $c_1(t)$ and $c_2(t)$ the spreading codes of respectively the first and the second user, with $b_i^{(j)}$ the bits of user i in the *j-th* period and $\hat{b}_i^{(j)}$ the partner's estimate. The parameters $\{a_{i,j}\}$ user i in the *j-th* period denote signal amplitude and so represent power allocation. Finally $X_1(t)$ and $X_1(t)$ are the the signals for each users. The decode-and-forward scheme can be represented as in Eq. (1.41)

$$\begin{aligned} X_1(t) &= a_{1,1}b_1^{(1)}c_1(t),\ a_{1,2}b_1^{(2)}c_1(t),\ a_{1,3}b_1^{(2)}c_1(t)+a_{1,4}\hat{b}_1^{(2)}c_1(t) \\ X_2(t) &= \underbrace{a_{2,1}b_2^{(1)}c_2(t),}_{Period_1}\ \underbrace{a_{2,2}b_2^{(1)}c_2(t),}_{Period_2}\ \underbrace{a_{1,4}\hat{b}_1^{(2)}c_1(t)+a_{2,4}b_2^{(2)}c_2(t)}_{Period_3} \end{aligned} \tag{1.41}$$

The first period is used to send information to the base station, the second one is used to send information both to the base station and the cooperative partner, finally in the last period after the decoding of previous signal, a new cooperative signal is constructed and sent to the base station. In this scheme parameters $\{a_{i,j}\}$ can be tuned but always a mean power constraint is maintained: when the interuser channel is favorable more power will be allocated to cooperation, when is not cooperation is reduced achieving an adaptability to channel conditions. For optimal decoding also the base station needs to know the error characteristics of the interuser channel.

1.3.3 Coded cooperation

Coded cooperation works by sending different portions of users word via different fading paths. To describe this typology we follow [12] and [13]. In a simple scheme users divide their source data into blocks, each one added with a CRC check. From these blocks a codeword of N bits is made. Each codeword is punctured into two segments, the first of length $N_1 < N$ in such a way that this new segment is a new weaker codeword while the other is the one of puncturing bits of length N_2. In the first temporal slot each source trasmits its own segment of length N_1, hears the interuser channel and attempt to decode the partner's codeword. If the attempt is successful or not can be verified by the users thanks to the CRC check. If it is successful the user can decode the codeword of the partner and send partner's second code partition of N_2 bits to the basestation in the next frame, viceversa if it is not the user sends its own puncturing segment. In both the cases, always N bits are sent and it is said that the level of cooperation between the two users is N_2/N_1. Notice that various general codes can be used and stronger the code better the performance.

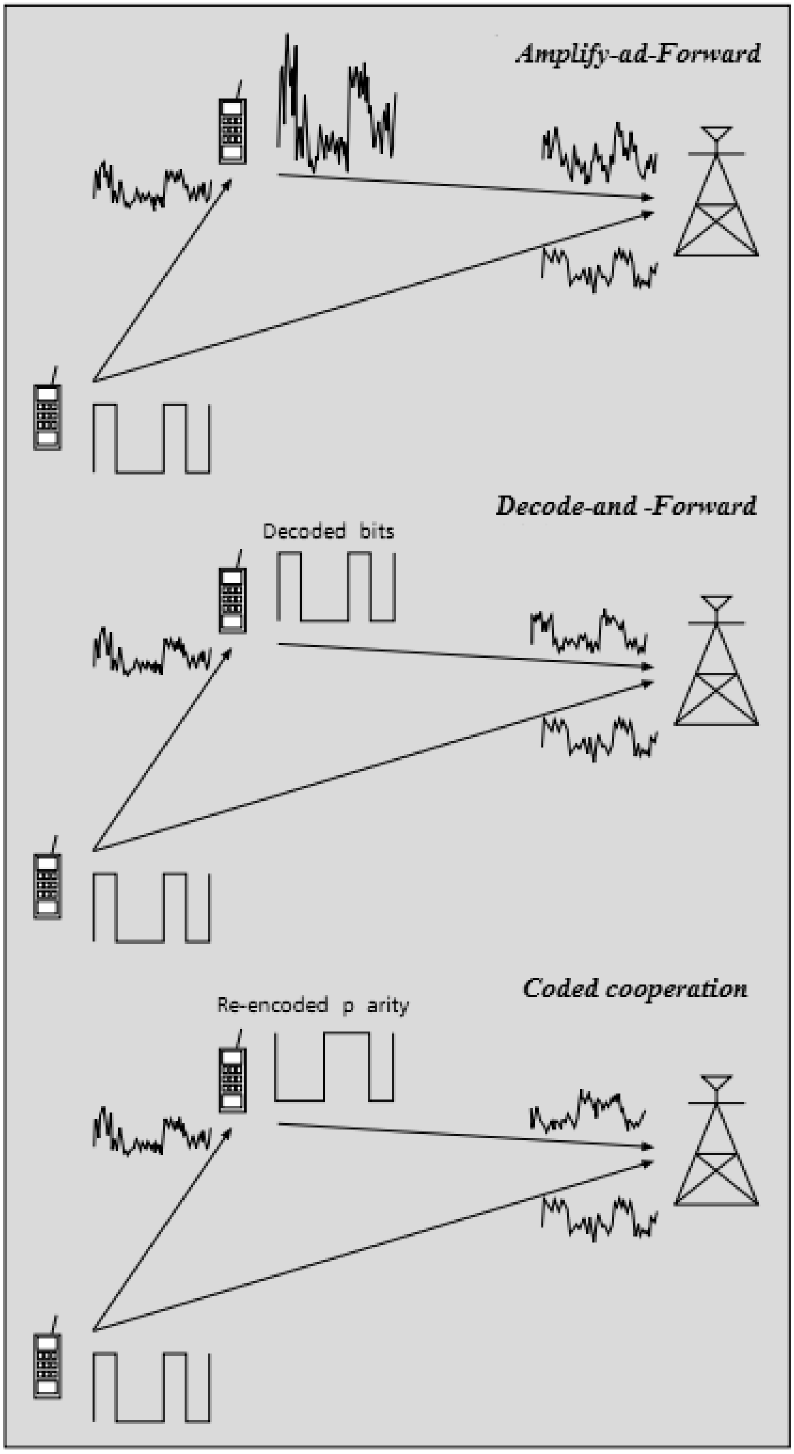

Figure 1.9: Comparison of different cooperative methods, with only user's cooperation

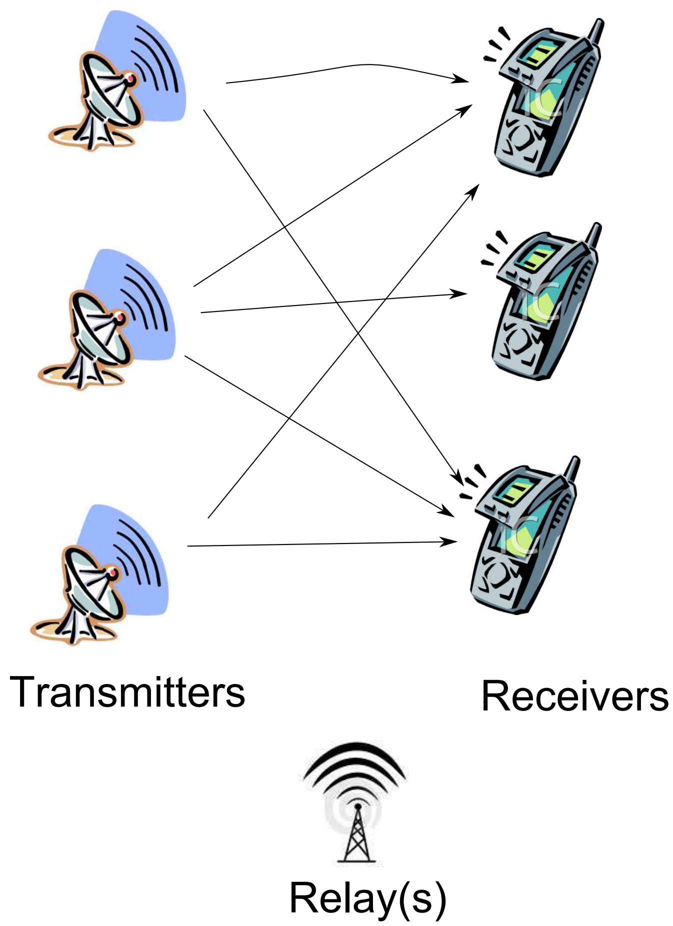

Figure 1.10: Network scenario

1.4 Problem statement

How could cooperation with relays improve the capacity and DoF of IA schemes? Imagine the presence in the network of a certain number of relays which have the aim of aiding the receivers to decode all the packets sent to them by the transmitters (see Fig. 1.10).

In [8] Cadambe and Jafar demonstrate that for a X-network aided by R relays with time-varying/frequency selective channel gains, the capacity improvement due to of cooperation is only up to a $o(\log(SNR))$. Any way the hypothesis made in [8, 3, 2] are not always realistic: the authors always speak about uncorrelated channels, high SNR and arbitrarily long temporal slots.

The underlying principle of this work is that relays could really be helpful, because it is possible that they could let the system better approximate Cadambe and Jafar's hypothesis, and as a consequence, to raise the DoFs of the system. For example relays could let a channel that results somehow correlated to become uncorrelated, or they could raise the SNR etc.

The basic idea is to understand whether scenarios, that can not reach perfect IA conditions, i.e. infinite SIR, could improve their performances thanks to the use of relays, raising their SIR or even better leading to an infinite SIR. Another question to understand could be the possibilities of cooperation on improving the convergence speed of IA algorithms.The investigation of the necessary number of relays, how many packets shall be retransmitted by the relays and with which scheme, how the noise impacts the system performance, how much the channel propagation is important in such a system etc. are all open problems that this work aims to solve.

Chapter 2

System models and algorithms

This chapter would like to give the reader all the foundations necessary to understand the work of this thesis. In the first section the aim is to introduce the scenarios on which the work is based on, explaining the topologies of the analyzed networks and the hypothesis on the system model. In the second section, all the algorithms of Interference Alignment, which have been implemented and used, are going to be listed and explained.

2.1 System model

As written before, the target of the thesis is to understand how cooperation could be useful for Interference Alignment. At first the model of the *K-network* introduced in [2] has to be briefly recalled. It is a network constituted of K transmitter-receiver couples where each transmitter wants to communicate only with its own receiver. Each transmitter could have several streams to transmit to its own receiver and so all the equipment is constituted by MIMO nodes. In [14] Jafar et al. demonstrated that there are some particular configurations on witch a perfect IA is achievable. These configurations can be derived by a really important formula, that could be called Jafar's inequality:

$$M_r + M_t - (K+1)S \geqq 0 \tag{2.1}$$

where like in previous chapter M_t and M_r are the number of transmitting and receiving antennas, S is the number of streams and K is the number of couples. If the left member is exactly zero it means that no more DoFs are available and so perfect IA is reached, with a throughput of $K \times S$ packets/slot[1].

Note that this formula is only valid for the so called "symmetric configurations", that is that type of configurations on which each couple has the same characteristics in number of receiving and transmitting antennas and number of streams. Note also that the formula is designed for only one temporal slot.

Starting from the *K-network*, in our job several network topologies are analyzed. All of them are constituted of K sub-networks and on each of them always only one receiver and only one transmitter are present, while a varying number of relays could be provided. Dealing with the hypothesis of [2], the transmitter and each relay belonging to one sub-network generate traffic only for the receiver belonging to that sub-network. Note that with this hypothesis, if no relay is present in all the network, a *K-network* is re-obtained.

On the following lines all the scenarios utilized are presented, each of them configured with the hypothesis already listed.

2.1.1 Scenario 1

This scenario is used in the first part of our job, examining the possibility of a capacity improvement in terms of throughput, in respect to the results already acquired by Jafar and alii in [14]. Starting from one *K-network* configuration (a synonym could be "standard configuration") we try to modify the scenario adding relays. As an example pick the simplest configuration, that in a short way could be called as (3c, 2a, 1sl, 1str) where *c* is the number of couples, *a* the number of antennas, *sl* the number of slot, *str* the number of stream, which identify the *K-network* with $K = 3$, $M_r = M_t = 2$, $S = 1$ (later on this notation will be always used also referring to other topologies).

[1] Pay attention that from here on the term "packet" is used to refer to each transmitted modulated symbol belonging to a specific packet. The terms are voluntarily confused to give a simpler representation in reader's mind.

We try to extend this topology from the point of view of the equipment used, adding relays, and from the temporal point of view, passing to a system designed for more than one slot. Finally a new couple is added. This new one could let the system achieve a better throughput if possible.

Delving more into the argument it is decided to add a fixed number of relays for the first K couples with a number of streams for each sub-network that is:

$$S_k = N_k + 1 \, k \in \{1, 2, ..., K\} \tag{2.2}$$

$$S_{K+1} = 1 \tag{2.3}$$

where N_k is the number of relays and the subscript k identify the k-th sub-network.

Each transmitter and all the relays belonging to one of the first K sub-network transmit one stream each. In this way relays behave more like other sources. The number of temporal slots of the whole system is equal to:

$$slot = S_k = N_k + 1 \, k \in \{1, 2, ..., K\} \tag{2.4}$$

Note that in our assumptions each source transmit the same frame in all the slots on which the system is configured.

The reader will understand that with this scenario, the topology of the network is modified: from the point of view of the single transmitter/relay the system can not be viewed like a *K-network* because the receivers have exactly a number of streams given by Eqs. (2.2), (2.3), while all the transmitters/relays have only one stream each. Moreover the system is not symmetric anymore because the last sub-system has different conditions. We decide to call each sub-network a *nV-network* where n is the number of relays. As an example of this topology, we introduce the most similar configuration to the previous one ((3c, 2a, 1sl, 1str)), which will be used also later on. It is a system with four sub-networks where the first three are *nV-networks* with one relay each, all the equipment has $M = M_t = M_r = 2$ antennas, the number of streams for the first three receivers is equal to two while for the last one is just one and finally the system works in two temporal slots. This example can be briefly called (3+1c, 2a, 2sl, 2str_rx) 1 relay per node. Note that in this scenario the number of streams per each transmitter/receiver is always equal to one but the number of streams at the receiver side could vary. Note also that the last sub-network (the one that let the system to have the possibility to achieve the throughput) has always one transmitter one receiver and only one stream.

Having described the system, we have also to imagine a model that could could efficiently describe the new scenario with relays and more than one slot. With the notation first introduced in Ch. 1, recalling the formulas and imagining the system working in just two slots, and so with only one relay per sub-network, we have that the received signal for the stream x_l is:

$$y_k^{(1)} = \sum_{k=1}^{K+1} H_{k,l} V_l^{(1)} \widetilde{x}_l + w_k \tag{2.5}$$

$$y_k^{(2)} = \sum_{k=1}^{K+1} H_{k,l} V_l^{(2)} \widetilde{x}_l + w_k \tag{2.6}$$

where $y_k^{(1)}$ and $y_k^{(2)}$ are the received signals of receiver k in the first and second slot respectively.

Since all streams are transmitted in both slots, it is possible to unify the structures in this way:

$$y_k = \begin{bmatrix} y_k^{(1)} \\ y_k^{(2)} \end{bmatrix} = \sum_{k=1}^{K+1} \begin{bmatrix} H_{k,l} & 0 \\ 0 & H_{k,l} \end{bmatrix} \begin{bmatrix} V_l^{(1)} \\ V_l^{(2)} \end{bmatrix} \widetilde{x}_l + w_k \tag{2.7}$$

As a consequence it is possible to define equivalent channel matrices, precoders and decoders: $\bar{H}_{k,l} = \begin{bmatrix} H_{k,l} & 0 \\ 0 & H_{k,l} \end{bmatrix}$ $\bar{V}_l = \begin{bmatrix} V_l^{(1)} \\ V_l^{(2)} \end{bmatrix}$ and $\bar{U}_k = \begin{bmatrix} U_k^{(1)} \\ U_k^{(2)} \end{bmatrix}$.

Including in the system the relays, they must have their own precoding vectors for they own streams respectevly (in both the slots):

$$x_l = \begin{bmatrix} V_{R_l}^{(1)} \\ V_{R_l}^{(2)} \end{bmatrix} \widetilde{x}_l \tag{2.8}$$

All the relays and the transmitters must have their own precoding vectors (see Ch. 1) for each slot with dimensions:

$$dim(V_j^{(i)}) = M \times 1 \tag{2.9}$$

Conversely for the receivers' decoding matrices:

$$dim(U_k^{(i)}) = S_k \times M \, k \in \{1, 2, ..., K\} \tag{2.10}$$

$$dim(U_{K+1}^{(i)}) = 1 \times M \tag{2.11}$$

Now putting all together, the system can be modeled by the formulas below:

$$\begin{aligned} y_k &= \begin{bmatrix} H_{k,k} & 0 \\ 0 & H_{k,k} \end{bmatrix} \begin{bmatrix} V_k^{(1,1)} \\ V_k^{(1,2)} \end{bmatrix} \widetilde{x}_k^{(1)} \\ &+ \begin{bmatrix} H_{k,R_k} & 0 \\ 0 & H_{k,R_k} \end{bmatrix} \begin{bmatrix} V_{R_k}^{(f+1,1)} \\ V_{R_k}^{(f+1,2)} \end{bmatrix} \widetilde{x}_k^{(2)} && (2.12) \\ &+ int_k + w_k && (2.13) \end{aligned}$$

$$\begin{aligned} int_k &= \sum_{l \neq k}^{K} (\begin{bmatrix} H_{k,l} & 0 \\ 0 & H_{k,l} \end{bmatrix} \begin{bmatrix} V_l^{(1,1)} \\ V_l^{(1,2)} \end{bmatrix} \widetilde{x}_l^{(1)} + \\ &+ \begin{bmatrix} H_{k,R_k} & 0 \\ 0 & H_{k,R_k} \end{bmatrix} \begin{bmatrix} V_{R_k^{(f)}}^{(2,1)} \\ V_{R_k}^{(2,2)} \end{bmatrix} \widetilde{x}_k^{(2)}) + \\ &+ \begin{bmatrix} H_{K+1,l} & 0 \\ 0 & H_{K+1,l} \end{bmatrix} \begin{bmatrix} V_{K+1}^{(1,1)} \\ V_{K+1}^{(1,2)} \end{bmatrix} \widetilde{x}_{K+1}^{(1)} && (2.14) \end{aligned}$$

where $k \in \{1, 2, ..., K\}$,while for the last receiver:

$$y_{K+1} = \begin{bmatrix} H_{K+1,K+1} & 0 \\ 0 & H_{K+1,K+1} \end{bmatrix} \begin{bmatrix} V_{K+1}^{(1,1)} \\ V_{K+1}^{(1,2)} \end{bmatrix} \widetilde{x}_{K+1}^{(1)} + int_{K+1} + w_{K+1} \tag{2.15}$$

$$
\begin{aligned}
int_{K+1} \quad = \quad & \sum_{l=1}^{K}\Big(\begin{bmatrix} H_{K+1,l} & 0 \\ 0 & H_{K+1,l} \end{bmatrix} \begin{bmatrix} V_l^{(1,1)} \\ V_l^{(1,2)} \end{bmatrix} \widetilde{x}_l^{(1)} + \\
& + \begin{bmatrix} H_{K+1,R_l} & 0 \\ 0 & H_{K+1,R_l} \end{bmatrix} \begin{bmatrix} V_{R_l}^{(2,1)} \\ V_{R_l}^{(2,2)} \end{bmatrix} \widetilde{x}_l^{(2)} \Big)
\end{aligned} \tag{2.16}
$$

Obviously in case of more than one slot the matrices become greater. In this scenario we can say that the total dimensions of the unified structures can be written like:

$$dim(\bar{H}_{k,l}) = (slot * M_r) \times (slot * M_t) \tag{2.17}$$

$$dim(\bar{V}_l) = (slot * M_t) \times 1 \tag{2.18}$$

$$dim(\bar{U}_k) = S_k \times (slot * M_r) \tag{2.19}$$

$$dim(\bar{C}_k) = (slot * M_r - S_k) \times (slot * M_r) \tag{2.20}$$

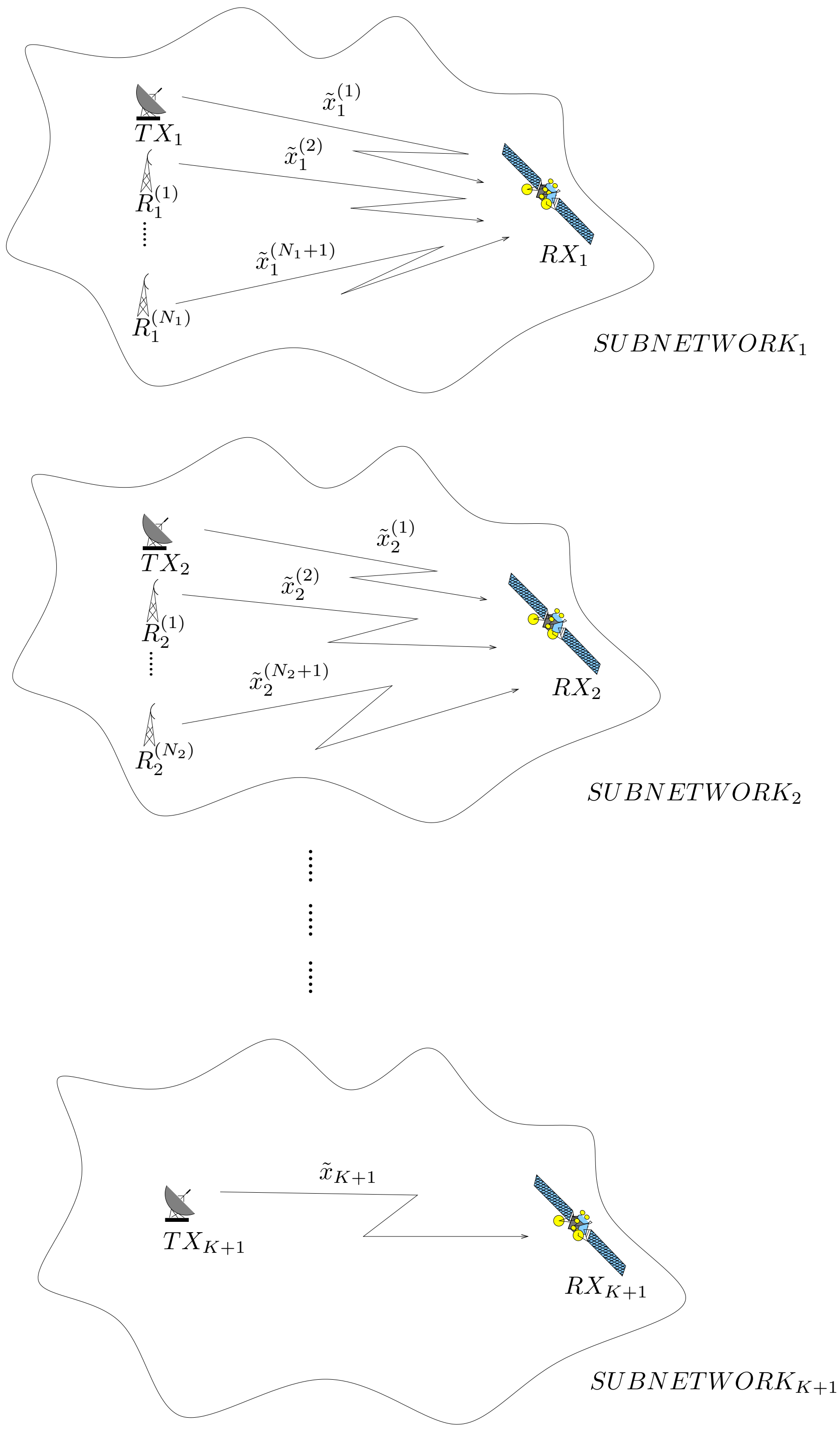

Figure 2.1: Scenario 1

2.1.2 Scenario 2

This new scenario is used to analyze convergence speed with cooperation. It is exactly the same to the one analyzed before except that it does not provide the extra sub-network (the one that could increase the throughput). In this way all sub-networks are *nV-networks* with the same characteristics in terms of antennas and streams. As before there is a transmitter, a receiver and some relays for each sub-network and each source transmits a packet in all the slots the system is configured. The equations that describe this model (in the case of a two slots system) are:

$$\begin{aligned} y_k &= \begin{bmatrix} H_{k,k} & 0 \\ 0 & H_{k,k} \end{bmatrix} \begin{bmatrix} V_k^{(1,1)} \\ V_k^{(1,2)} \end{bmatrix} \widetilde{x}_k^{(1)} \\ &+ \begin{bmatrix} H_{k,R_k} & 0 \\ 0 & H_{k,R_k} \end{bmatrix} \begin{bmatrix} V_{R_k}^{(2,1)} \\ V_{R_k}^{(2,2)} \end{bmatrix} \widetilde{x}_k^{(f+1)} \\ &+ int_k + w_k \end{aligned} \tag{2.21}$$

$$\begin{aligned} int_k &= \sum_{l \neq k}^{K} \Big(\begin{bmatrix} H_{k,l} & 0 \\ 0 & H_{k,l} \end{bmatrix} \begin{bmatrix} V_l^{(1,1)} \\ V_l^{(1,2)} \end{bmatrix} \widetilde{x}_l^{(1)} + \\ &+ \begin{bmatrix} H_{k,R_k^{(f)}} & 0 \\ 0 & H_{k,R_k^{(f)}} \end{bmatrix} \begin{bmatrix} V_{R_k}^{(2,1)} \\ V_{R_k}^{(2,2)} \end{bmatrix} \widetilde{x}_k^{(f+1)} \Big) \end{aligned} \tag{2.22}$$

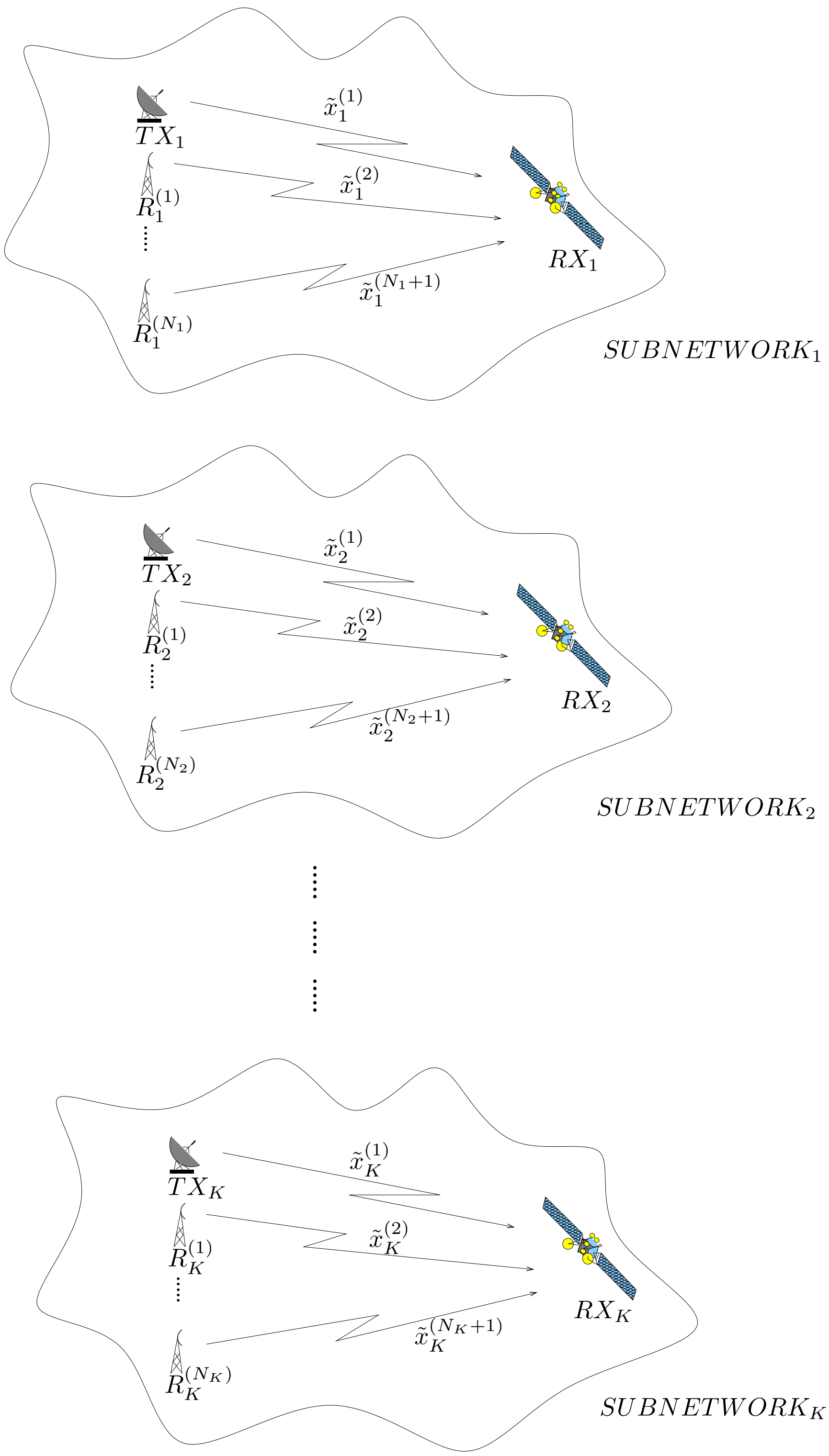

Figure 2.2: Scenario 2

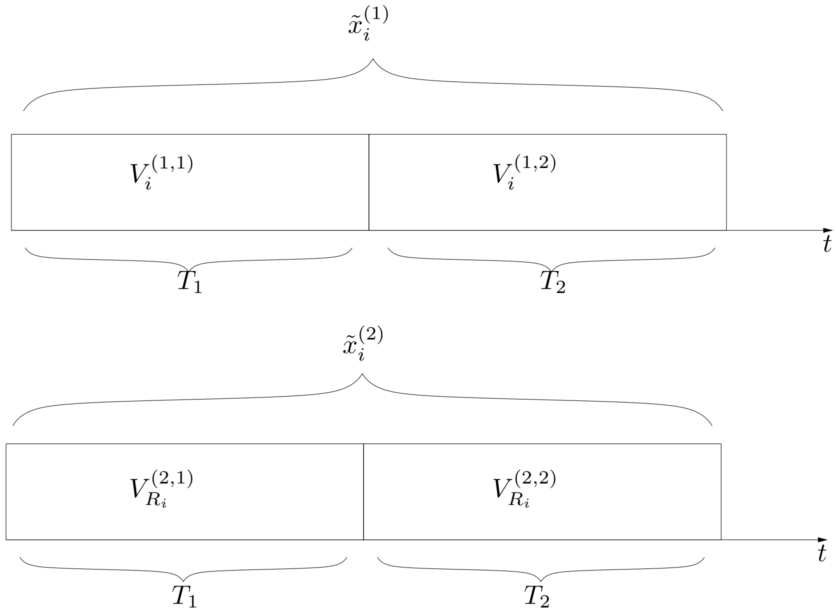

Figure 2.3: Temporal view for Scenario 1 and Scenario 2

2.1.3 Scenario 3

This last scenario is used for the analysis of both convergence and capacity improvement. The topology is the same as in Scenario 2: there are several sub-networks with one transmitter, one receiver and a fixed number of relays. The difference is that in this scenario every source transmits the *same packet* in all the slots (and so in each sub-network there is only one stream). The very important hypothesis made here is that all the sources have the knowledge of all the channels of the same sub-network. Obviously this could be a hard hypothesis but might be verified in a satellite context, because it is possible the receiver sends information to the sources about the channels.

Focusing in just one slot the system could be seen as:

$$y_k = H_{k,k} V_k \widetilde{x}_k + \sum_{f=1}^{N_k} H_{k,R_k} V_{R_k^{(f)}} \widetilde{x}_k + int_k + w_k \tag{2.23}$$

where *int* is used as the sum of all the interference, such as:

$$\begin{aligned} int_k &= \sum_{l \neq k}^{K} \left(H_{k,l} V_l \widetilde{x}_l + \sum_{f=1}^{N_k} H_{k,R_l} V_{R_l^{(f)}} \widetilde{x}_l \right) \\ &= \sum_{l \neq k}^{K} \left(H_{k,l} V_l + \sum_{f=1}^{N_k} H_{k,R_l} V_{R_l^{(f)}} \right) \widetilde{x}_l \end{aligned} \tag{2.24}$$

This model can also be rewritten in a unified manner: as the stream is the same, the matrices of channels and the precoders can be merged in single structures for the same sub-network

$$\bar{H}_{k,l} = \begin{bmatrix} H_{k,l} & H_{k,R_l^{(1)}} & \dots & H_{k,R_l^{(N_k)}} \end{bmatrix} \tag{2.25}$$

$$\bar{V}_l = \begin{bmatrix} V_l \\ V_{R_l^{(1)}} \\ \vdots \\ V_{R_l^{(N_k)}} \end{bmatrix} \tag{2.26}$$

Notice that to do this it is necessary to exploit the important hypothesis made before on the mutual knowledge of the channel by all the sources.

In this way Eq. (2.23) and Eq. (2.24) can be rewritten in the following way:

$$y_k = \bar{H}_{k,k}\bar{V}_k\widetilde{x}_k + int_k + w_k \tag{2.27}$$

$$int_k = \sum_{l \neq k}^{K} \bar{H}_{k,l}\bar{V}_l\widetilde{x}_l \tag{2.28}$$

Trying to extend the system for more than one slot, we imagined a scenario in which there are sources transmitting the same packet in all the slots in the same sub-network. As a consequence in the example of two slots the model could be described in the following way:

$$y_k = \begin{bmatrix} y_k^{(1)} \\ y_k^{(2)} \end{bmatrix} = \begin{bmatrix} H_{k,l} & H_{k,R_l^{(1)}} & \dots & H_{k,R_l^{(N_k)}} \\ H_{k,l} & H_{k,R_l^{(1)}} & \dots & H_{k,R_l^{(N_k)}} \end{bmatrix} \begin{bmatrix} V_l \\ V_{R_l^{(1)}} \\ \vdots \\ V_{R_l^{(N_k)}} \end{bmatrix} \widetilde{x}_l + int_k + w_k \tag{2.29}$$

So the channel matrices become:

$$\bar{H}_{k,l} = \begin{bmatrix} H_{k,l} & H_{k,R_l^{(1)}} & \dots & H_{k,R_l^{(N_k)}} \\ H_{k,l} & H_{k,R_l^{(1)}} & \dots & H_{k,R_l^{(N_k)}} \end{bmatrix} \tag{2.30}$$

while precoders do not vary.

In this model the precoders remain exactly the same for all the slots. This is an important difference with the previous scenarios. Focusing finally on the decoders, they have always just one stream.

In order to complete the description, let us now have a look at the structure dimensions in this scenario:

$$dim(H_{k,l}) = (slot * M_r) \times ((N_k + 1) * M_t) \tag{2.31}$$

$$dim(V_l) = ((N_k + 1) * M_t) \times 1 \tag{2.32}$$

$$dim(U_k) = 1 \times (slot * M_r) \tag{2.33}$$

$$dim(C_k) = (slot * M_r - 1) \times (slot * M_r) \tag{2.34}$$

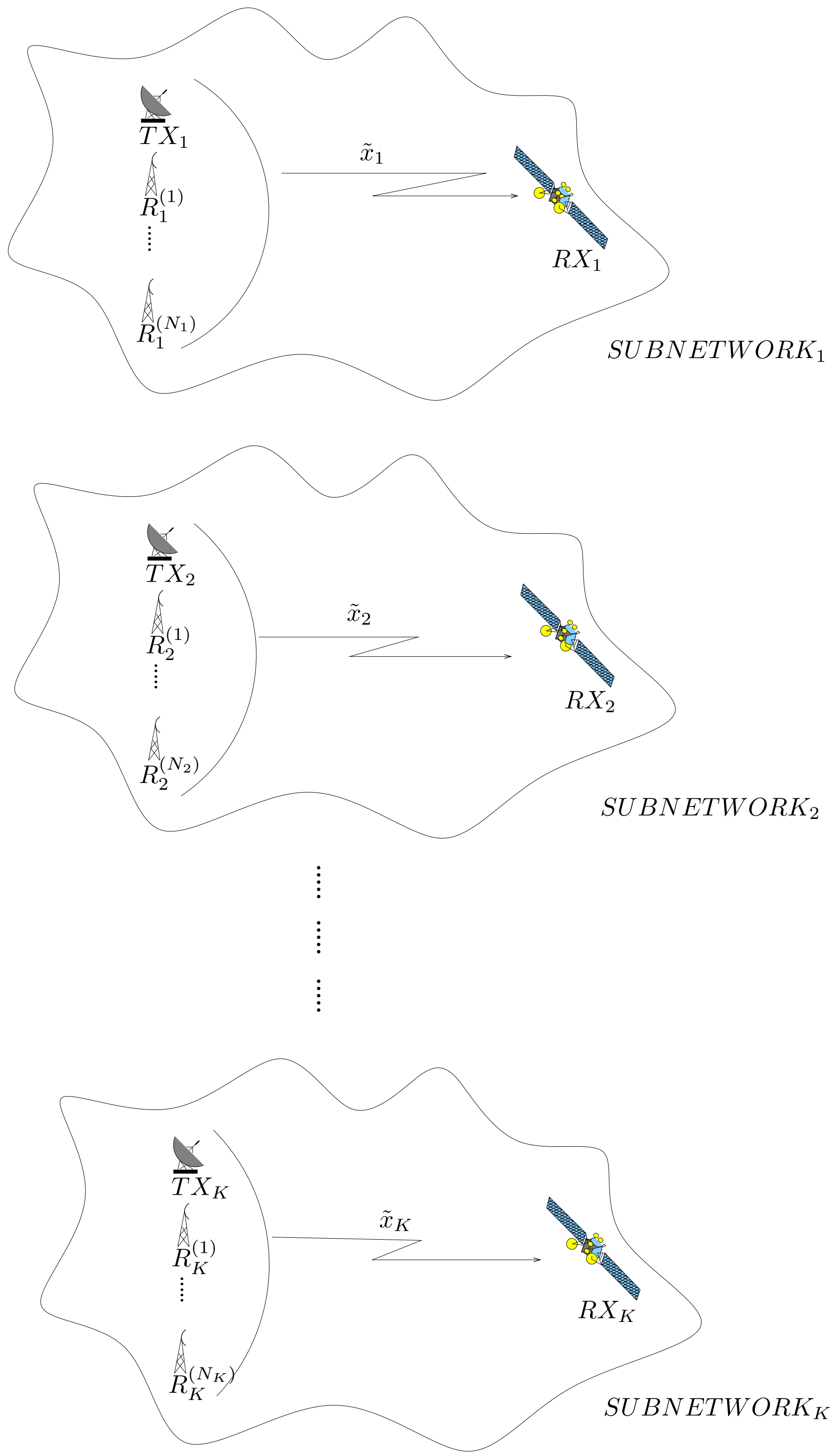

Figure 2.4: Scenario 3

2.2 Algorithms for cooperative IA

To simulate the scenarios explained before, we create some Matlab scripts that exploit the "unified system matrices" introduced in the last section. Obviously they differe depending on the scenario on which the system is working.

We try different algorithms of IA adapted for a cooperative context. The first two of them are a review of some explained in Ch. 2, while the third is a new one.

To compare the configurations and the algorithms we analyze three metrics, that are listed here:

- The residual interference on the subspace of useful signal, in formulas:

$$\sum_{k=1}^{K}\sum_{l\neq k}^{K}||H_{k,l}V_l - C_kC_k^H H_{k,l}V_l||^2$$

  as from [4]. Adapting this metric to our cooperative case, the formula develops in:

$$\begin{aligned} INT_{res} = & \sum_{k=1}^{K}\sum_{l\neq k}^{K}||H_{k,l}V_l - C_kC_k^H H_{k,l}V_l||^2 + \\ & + \sum_{k=1}^{K}\sum_{l\neq k}^{K}\sum_{f=1}^{N_l}||H_{k,R_l^{(f)}}V_{R_l^{(f)}} - C_kC_k^H H_{k,R_l^{(f)}}V_{R_l^{(f)}}||^2 \end{aligned} \quad (2.35)$$

  where the first term is produced by transmitters and the second one is produced by relays.

- The sum of SINR seen at each receiver where

$$SINR_k = \frac{U_k^H H_{k,k}V_kV_k^H H_{k,k}^H U_k + \sum_{f=1}^{N_k} U_k^H H_{k,R_k^{(f)}} V_{R_k^{(f)}} V_{R_k^{(f)}}^H H_{k,R_k^{(f)}}^H U_k}{\sum_{l\neq k}^{K}\left(U_k^H H_{k,l}V_lV_l^H H_{k,l}^H U_k + \sum_{f=1}^{N_l} U_k^H H_{k,R_l^{(f)}} V_{R_l^{(f)}} V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H U_k\right) + N_0 I} \quad (2.36)$$

  $SINR_k$ is the SINR at the k-th receiver. As a consequence our metric will be simply: $\sum_{k=1}^{K} SINR_k$.

- The spectral efficiency calculated as: $\eta = \log_2(1 + SINR)$.

Finally, to better understand how each configuration behaves, we introduce two "diagnostic" parameters:

- *Failure Rate* is the percentage of total realizations that cannot reach the threshold of 80 dB SINR.

- *Fake Rate* is the percentage of total realizations that can reach the 80 dB SINR but then goes down. We say that a configuration that has a fake rate much grater than zero suffers of instability.

2.2.1 "Zero-Forcing" algorithm

Adapting what explained in Ch. 2 and in [4], our the Zero Force Algorithm aims to minimize the function (2.35).

As before V_l are computed by choosing the least dominant eigenvector of:

$$\sum_{k \neq l}^{K} H_{k,l}^H (I - C_k C_k^H) H_{k,l} \tag{2.37}$$

while $V_{R_l^{(f)}}$ are computed by choosing the eigenvector associated to the least dominant eigenvalue of :

$$\sum_{k \neq l}^{K} H_{k,R_l^{(f)}}^H (I - C_k C_k^H) H_{k,R_l^{(f)}} \tag{2.38}$$

Notice that thanks to the hypothesis done, the number of streams of each precoders is always equal to one that is why we are always taking only the eigenvector associated to the least dominant eigenvalue.

Matrices C_k are computed by choosing the $dim_{col}(C_k) - S_k$ dominant eigenvectors of the following matrix:

$$\sum_{l \neq k}^{K} H_{k,l} V_l V_l^H H_{k,l}^H + \sum_{l \neq k}^{K} \sum_{f=1}^{N_k} H_{k,R_l^{(f)}} V_{R_l^{(f)}} V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H \tag{2.39}$$

As written in previous chapter the matrices C_k are the orthogonal complement of matrices U_k. In this way, having calculated all the C_k matrices and aiming to pass to the U_k matrices, we perform the subsequent steps:

1. $Q_k = \text{qr}\,(C_k)$, performing a QR decomposition of matrices C_k
2. fix the U_k matrices as the last S_k columns of Q_k

In the Appendix, the reader can find a better explanation of the passages before.

2.2.2 "Max-SINR" algorithm

This algorithm exploits the same idea of the one presented in Ch. 1 but it makes no assumption of reciprocity.

It generates the precoders in the same way of the previous algorithm, and so precoding matrices at each step are the same as the one before, while decoders are different.

Rewriting $SINR_k$ of Eq. (2.36) in the following way:

$$\frac{U_k^H \left(H_{k,k} F_k F_k^H H_{k,k}^H + \sum_{f=1}^{N_k} H_{k,R_k^{(f)}} V_{R_k^{(f)}} V_{R_k^{(f)}}^H H_{k,R_k^{(f)}}^H \right) U_k}{U_k^H \left[\sum_{l \neq k}^{K} \left(H_{k,l} F_l F_l^H H_{k,l}^H + \sum_{f=1}^{N_l} H_{k,R_l^{(f)}} V_{R_l^{(f)}} V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H \right) + N_0 I \right] U_k} \tag{2.40}$$

and calling

$$A_k = \left(H_{k,k} V_k V_k^H H_{k,k}^H + \sum_{f=1}^{N_k} H_{k,R_k^{(f)}} V_{R_k^{(f)}} V_{R_k^{(f)}}^H H_{k,R_k^{(f)}}^H \right) \tag{2.41}$$

$$M_k = \left[\sum_{l \neq k}^{K} \left(H_{k,l} V_l V_l^H H_{k,l}^H + \sum_{f=1}^{N_l} H_{k,R_l^{(f)}} V_{R_l^{(f)}} V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H \right) + N_0 I \right] \tag{2.42}$$

then

$$\begin{aligned} \max(SINR_k) \quad \Rightarrow \quad & \max\left(U_k^H A_k U_k \right) \\ & \text{sub} \left(U_k^H M_k U_k \right) = \text{c} \end{aligned} \tag{2.43}$$

The solution of this problem is well known in linear algebra: U_k is composed by the S_k dominant generalised eigenvectors of this problem: $A_k U_k = \lambda M_k U_k$.

In this way both precoders and decoders are optimized at each iteration.

2.2.3 Gradient descend algorithm

Looking at our target just as a minimization or a maximization problem, we understand that a gradient descend technique could be used.

Suppose *f(x)* is a function of the real-value column vector *x*. Gradient descend is an iterative algorithm which aims to find the optimum value of *x*, that minimizes the function *f*, in the following way:

$$x_{n+1} = x_n - \alpha \nabla f(x_n) \tag{2.44}$$

where x_{n+1} is vector x at iteration n+1, x_n is vector x at step n and operator ∇ is the gradient operator.

Picking Eq. (2.35), we try to minimize it using this technique. Note that the residual interference can be seen as:

$$INT_{res} = \sum_{k=1}^{K} \sum_{l \neq k}^{K} \left[\operatorname{tr}\left(V_l^H H_{k,l}^H U_k U_k^H H_{k,l} V_l\right) + \sum_{f=1}^{N_l} \operatorname{tr}\left(V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H U_k U_k^H H_{k,R_l^{(f)}} V_{R_l^{(f)}}\right)\right] \tag{2.45}$$

where tr is the trace operator. As a consequence INT_{res} is our f while our x is made by the union of all the precoders (for both the transmitters and the relay) and the decoders. Notice that our x is composed of matrices of complex values. The derivation is possible thanks to the complex gradient and the trace operator (cfr. Appendix) and the result is the following:

$$\nabla INT_{res} = \begin{bmatrix} \sum_{k\neq 1}^{K} H_{k,1}^{H} U_k U_k^H H_{k,1} V_1 \\ \vdots \\ \sum_{k\neq K}^{K} H_{k,K}^{H} U_k U_k^H H_{k,K} V_K \\ \sum_{k\neq 1}^{K} H_{k,R_1^{(f)}}^{H} U_k U_k^H H_{k,R_1^{(f)}} V_{R_1^{(f)}} \\ \vdots \\ \sum_{k\neq K}^{K} H_{k,R_K^{(f)}}^{H} U_k U_k^H H_{k,R_K^{(f)}} V_{R_K^{(f)}} \\ \sum_{l\neq 1}^{K} \left(H_{k,l} V_l V_l^H H_{k,l}^H + \sum_{f=1}^{N_l} H_{k,R_l^{(f)}} V_{R_l^{(f)}} V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H \right) U_1 \\ \vdots \\ \sum_{l\neq K}^{K} \left(H_{k,K} V_K V_K^H H_{k,K}^H + \sum_{f=1}^{N_l} H_{k,R_K^{(f)}} V_{R_K^{(f)}} V_{R_K^{(f)}}^H H_{k,R_K^{(f)}}^H \right) U_K \end{bmatrix} \tag{2.46}$$

After having found the new value of each matrix, we need to perform another step which is not directly correlated to the gradient descend algorithm but to the nature of our problem and object function. All the matrices must be *unitary*, but the passages before do not imply this condition. That is why we have to force it by "normalizing" the matrices. To do so, we perform the subsequent steps for every precoding and decoding matrix generically called B_k:

1. $Q_k = \text{qr}(B_k)$, performing a QR decomposition of matrices B_k
2. fix the B_k matrices as the first l columns of Q_k where l depends on the matrix analyzed.

If it is true that the matrices are obviously different from the ones found by gradient descend, it is also true that we are looking at solutions that stay in the Grassman's manifold (not in Stein's manifols). In other words with the two steps performed before, we are picking another solution, different from the one found by gradient descend, but which stays on the same vector space and this enables the algorithm to find the solution anyway. This the concept of Grassman's manifold which differs from the one of Stein's manifold on which the correct solution is only the peculiar solution.

Notice that these steps were not necessary in the algorithms presented in the last two subsections, because matrices at each step were built thanks to eigenvectors and so they resulted unitary.

We suppose the gradient step α is fixed at a heuristic value that will be empirically found. A derivation of the optimum fixed step is done in [15], but can not be applied to our case because the process of normalization at each steps invalidates this derivation.

Any way α can also be optimized and changed at each iteration as it could reach the convergence faster. On this we will focus during the next chapters.

Chapter 3

Capacity improvement analysis

As a first topic of our work we try to improve the capacity of the system exploiting cooperation. Is it possible to achieve an increment of throughput thanks to the use of cooperation? To answer this question, after a first analysis in various network topologies, Scenario 1 and Scenario 3 are exploited and the the results are given in next sections.

3.1 First analysis

At first, we try to imagine a network topology with four couples transmitter-receiver, with only one relay is associated to the last one. The system works on two temporal slots and the decoding is performed after the second one. In this context, there are no unified structures and so the first three transmitters have two different precoders, one for the first slot $V_{i,1}$ and one for the second slot $V_{i,2}$. The last transmitter has only one precoder, used in the first slot, while in the second slot the relay is exploited using its own precoder V_R. The first three transmitters send two different packets during the slots, while the last couple retransmits thanks to the relay that acts in a decode-and-forward way. All the receivers have their own two different decoders $U_{i,1}$, $U_{i,2}$. The calculation of the SINR is done imagining to have seven different receivers in the following way:

$$SINR_{i,1} = \frac{||U_{i,1}^H H_{i,i} V_{i,1}||^2}{\sum_{j \neq i}^4 ||U_{i,1}^H H_{i,i} V_{j,1}||^2} \tag{3.1}$$

$$SINR_{i,2} = \frac{||U_{i,2}^H H_{i,i} V_{i,2}||^2}{\sum_{j \neq i}^4 ||U_{i,2}^H H_{i,i} V_{j,2}||^2} \tag{3.2}$$

$$SINR_4 = \frac{||U_{4,1}^H H_{4,4} V_{4,1}||^2 + ||U_{4,2}^H H_{4,R} V_R||^2}{\sum_{j=1}^3 ||U_{i,1}^H H_{i,i} V_{j,1}||^2 + \sum_{j=1}^3 ||U_{i,2}^H H_{i,i} V_{j,2}||^2} \tag{3.3}$$

where $SINR_{i,1}$ and $SINR_{i,2}$ are the SINR of the first three receiver respectively in the first and second slot, while $SINR_4$ is the SINR for the last one. Empirical results show that perfect IA is not reachable.

We try an another scenario with unified structures. The topology is the same, more than one relay can be present in the network but only one associated to each sub-network. Each packet is transmitted or retransmitted in both the slots. The first three sub-networks have two packets each, while the last one has only one stream. Channel matrices have the form:

$$\bar{H}_{i,j} = \begin{cases} \begin{bmatrix} H_{i,j} & 0 \\ 0 & H_{i,j} \end{bmatrix} & subnetwork\, with\, no\, relay \\ \begin{bmatrix} H_{i,j} & 0 \\ 0 & H_{i,R_J} \end{bmatrix} & subnetwork\, with\, relay \end{cases} \tag{3.4}$$

while overall precoders can be figured as follows:

$$\bar{V}_i = \begin{cases} \begin{bmatrix} V_i^{(1)} \\ V_i^{(2)} \end{bmatrix} & subnetwork\, with\, no\, relay \\ \begin{bmatrix} V_i^{(1)} \\ V_{R_i}^{(2)} \end{bmatrix} & subnetwork\, with\, relay \end{cases} \tag{3.5}$$

Also in this way perfect IA can not be achieved.

3.2 Scenario 1 analysis

We exploit Scenario 1 (cfr. Ch. 2) focusing at first on two configurations: (3+1c, 2a, 2sl, 2str_rx) *3V-network*, 1 relay per transmission node and (5+1c, 2a, 2sl, 2str_rx) *5V-network*, 1 relay per transmission node. These configurations are listed in Table 3.1. As introduced in the last chapter, each *V-sub-network* is equipped with one relay: the transmitter sends a first stream while the relay sends the second one. Both entities transmit their own frame in both the slots. In the extra couple, the transmitter just sends its own frame in both the slots. The first K receivers have two streams to decode, while the last receiver just one.

To simulate these configurations, we exploit at the beginning only the first two algorithms (*Zero-Forcing* and *Max-SINR*).

Unfortunately our results are not so easy to interpret. We simulate many channel realizations, but not always the same behavior is obtained. The following Table lists the results obtained thanks to *Max-SINR* and it can clarify what stated above:

Table 3.1: Analysis on capacity improvement on Scenario 1

configuration	*SINR*	*Residual Interf.*	*Failure Rate*	*Fake Rate*	*N*[1]	*Spectral Efficiency*
((3+1)c, 2a, 2sl, 2str_rx) 3V-network	80dB	1	22%	50%	//	15
((3+1)c, 4a, 1sl, 2str_rx) 3V-network	inf	0	0%	0%	1600	122

[1]N is the number of iterations to reach 80 dB SINR

((5+1)c, 4a, 1sl, 2str_rx) 5V-network	72dB	10	77%	14%	//	65
((5+1)c, 4a, 1sl, 2str_rx) 5V-network	inf	0	0%	0%	6000	160

It is said that this sort of configurations, listed in the odd rows of the Table 3.1, suffers of instability problems: an high *fake-Rate* and *fail-Rate* for these two configurations are detected while only few realizations present a perfect IA behavior.

Moreover on these configurations the graphs of the algorithms used do not coincide anymore (*Zero-Forcing* algorithm seems to give always perfect IA) and so we cannot infer that these two configurations give perfect IA.

To better understand this kind of configurations we exploit the Gradient-Descend algorithm. It is less efficient because it takes a really high number of iterations (the step is not optimized), but it leads to results that are very similar to the ones obtained thanks to the *Max-SINR* and listed in Table 3.1. As a consequence, this analysis is taken into account.[2]

From all these considerations, we are obliged to say that these two configurations can not give perfect IA. To have a deeper insight of the problem, we try to analyze configurations that are similar from a computational point of view, not from a logic one. These configurations are presented in the even rows of Table 3.1 and they have the same matricial structures of the ones analyzed, thanks to the double number of antennas and the half number of slots, with the only difference that the channel matrix is not block diagonal, like in the configurations of interest, but they are full matrices. Modifying

[2]We think that these results are more reliable also because a great simulation campaign is done and *Max-SINR* algorithm seems to be more stable than *Zero-Forcing* algorithm in some critical configuration as the ones reported here.

the example of Eq. (2.12) we obtain:

$$\begin{aligned} y_k &= \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_k^{(1,1)} \\ V_k^{(1,2)} \end{bmatrix} \widetilde{x}_k^{(1)} \\ &+ \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix} \begin{bmatrix} V_{R_k^{(f)}}^{(2,1)} \\ V_{R_k^{(f)}}^{(2,2)} \end{bmatrix} \widetilde{x}_k^{(2)} \\ &+ int_k + w_k \end{aligned} \tag{3.6}$$

where sub-matrices *A, B, C, D* and *A', B', C', D'* are completely stochastic matrices.

On this kind of configurations perfect IA is reached, but with a very high number of iterations, which demonstrates that DoFs are most of all saturated.

It is our opinion that the structures of channel matrices in the cases of interest cannot lead to results that could be achievable thanks to completely full random matrices, because the correlation of those matrices in the first case can not be null as in the second one and this depends on the scenario and the topology of the network chosen. Notice otherwise that the configurations listed in the even rows can not lead to a capacity improvement because they exploit a double number of antennas which increases the degrees of freedom.

To conclude, the results described for configurations exploiting Scenario 1 can not allow us to say anything about the possibility to achieve a throughput increment.

In Fig. 3.2 the performances of SINR and residual interference in the space of useful signal are reported for the configuration with $K = 3+1$ listed in the first row of Table 3.1: reader can observe how the SINR for the block-channel configuration cannot reach the 80 dB and how much the residual interference is different from 0.

Analyzing similar configurations with higher K and higher number of relays for each sub-network, similar results are obtained and anything new can been added.

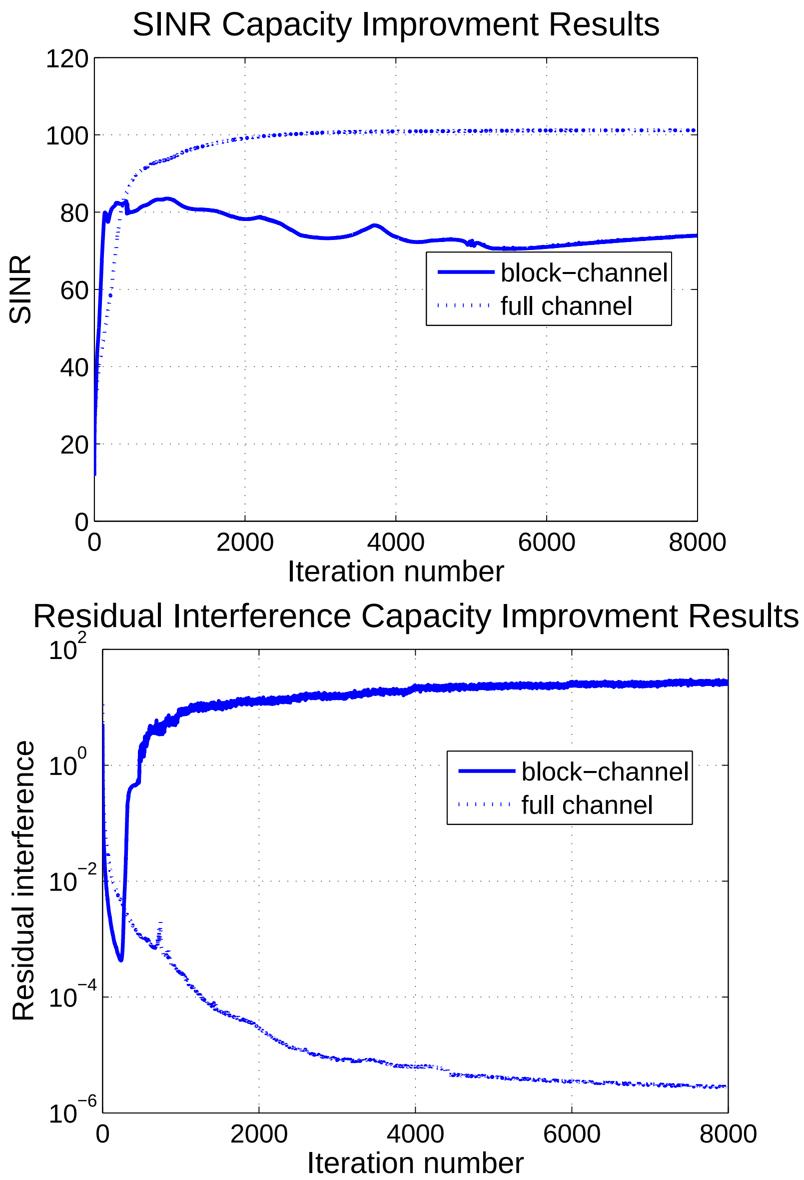

Figure 3.1: Differences between block-channel matrices and full-channel matrices on capacity improvement

3.3 Scenario 3 analysis

Aware of the results listed before, we try different ways. In the specific, we would like to understand if exploiting Scenario 3 a throughput increment is achievable.

3.3.1 One slot analysis

We choose configurations which are similar to the "classic" critical configuration (3c, 2a, 1sl, 1str) trying to see if a throughput of more than 3 pck/slot is achievable. In the specific all the configurations reported in Table 3.2 deal with Scenario 3 and all the three algorithms lead to same results:

Table 3.2: Analysis on capacity improvement on Scenario 3

configuration	*SINR*	*Residual Interf.*	*Failure Rate*	*Fake Rate*
(3c, 2a, 1sl, 1str_rx) 3V-network-1rly each	inf	0	0%	0%
(3c, 2a, 1sl, 1str_rx) 3V-network-2rly each	inf	0	0%	0%
(4c, 2a, 1sl, 1str_rx) 4V-network-1rly each	inf	0	0%	0%
(4c, 2a, 1sl, 1str_rx) 4V-network-2rly each	inf	0	0%	0%

(5c, 2a, 1sl, 1str_rx) 5V-network-1rly each	inf	0	0%	0%
(5c, 2a, 1sl, 1str_rx) 5V-network-2rly each	inf	0	0%	0%
(6c, 2a, 1sl, 1str_rx) 6V-network-1rly each	30 dB	1	100%	0%
(6c, 2a, 1sl, 1str_rx) 6V-network-2rly each	inf	0	0%	0%

As the reader can see, all the configurations except one give a perfect IA with a throughput equal or higher than 3 pck/slot (performed by the critical configuration). At first glance the results can induce the reader to think that an effective capacity improvement can be realized. Nevertheless a more global view of the system is necessary. At first Jafar's inequality, presented in [14] and which gives the feasibility of IA in symmetric systems, has to be recalled:

$$M_r + M_t - (K+1)S_k \geqq 0 \tag{3.7}$$

We look at Scenario 3 trying to apply the inequality to the model. Actually this is possible because all the entities, belonging to the same sub-network, transmit the same stream and thanks to the important hypothesis made on mutual knowledge of the sources, the model can be interpreted as at the transmitter side a unique entity would be present with

$$\bar{M_t} = (N_k + 1) * M \tag{3.8}$$

antennas, while at the receiver side the only one entity is just the receiver which has only M antennas. This is also perfectly comprehensible from the mathematical model of Scenario 3 presented in the last chapter.

In this way the scenario could be viewed in an unified manner and a *K-network* is re-obtained where the (global) transmitter has $\bar{M}_t$ antennas, while the receiver has just $\bar{M}_r = M$ antennas. As written above the stream is just one for each sub-network and K is the number of sub-networks. Looking at the system in this simplified way, Jafar's inequality can be reused and the results listed in the Table deal with this view becoming now more comprehensible.

Finally we can say that thanks to the hypothesis made for this scenario a capacity improvement of at most 5 pck/slot with one relay each sub-network is achievable because we reach conditions that can lead our system to have more degrees of freedom than the "classic" configuration, but this new configuration is well-known in literature.

3.3.2 More slots analysis

Empirical results and correlation problems

Now the question could be: aware of these conditions and the hypotheses made, is it possible to have an higher increment of throughput just decoding after more than one slot? To answer this question it is useful to exemplify first. Let us focus on the configuration (3+1c, 2a, 2sl, [2, 2, 2, 1]str) *4V-network*, similar to the configuration presented in the previous subsection but with a different scenario. Each sub-network has its own relay associated to the transmitter and the two sources transmit different packets but the same frame in both the slots. The system model of this configuration has the following feature:

$$y_k = \bar{H}_{k,k}\bar{V}_k^{(1)}\widetilde{x}_k^{(1)} + \bar{H}_{k,k}\bar{V}_k^{(2)}\widetilde{x}_k^{(2)} + int \tag{3.9}$$

where $y_k = \begin{bmatrix} y_k^{(1)} \\ y_k^{(2)} \end{bmatrix}$, $\bar{H}_{k,k} = \begin{bmatrix} H_{k,k} & H_{k,R_k} \\ H_{k,k} & H_{k,R_k} \end{bmatrix}$ and $\bar{V}_k^{(i)} = \begin{bmatrix} V_k^{(i)} \\ V_{R_k}^{(i)} \end{bmatrix}$.

As a consequence we see that the received signal is made of two parts one for each slot and has dimensions $2M_r \times 1$. The precoder vector has dimensions $(N_k + 1)M_t \times 1$ and the channel has dimensions $2M_r \times (N_k + 1)M_t$. Empirical results made thanks to our simulator and exploiting all the algorithms presented in the last chapter show that this kind of configurations can not achieve perfect IA. Aware of the conclusions found before in last subsection, we understand that correlation of the channel ($y_k^{(1)}$and $y_k^{(2)}$ are the same) is the reason that can not lead to good results. This is not a great problem because we find that just modifying channel matrices f.e. by inserting some zero blocks (do not letting some device to transmit for some stream), it is possible to reach the convergence of the system.

In this way not all channel matrices will be like $\bar{H}_{k,k}$ but some of them will have, for example, the subsequent form:

$$\hat{H}_{k,k} = \begin{bmatrix} H_{k,k} & 0 \\ H_{k,k} & H_{k,R_k} \end{bmatrix} \tag{3.10}$$

Other configurations that can be analyzed are low rate networks. Let's choose as an example a configuration (7c, 2a, 2sl, 1str) *7V-network*, with one relay associated to each transmitter. The model for this configuration can be:

$$y_k = \bar{H}_{k,k}\bar{V}_k^{(1)}\widetilde{x}_k^{(1)} + int \tag{3.11}$$

where $y_k = \begin{bmatrix} y_k^{(1)} \\ y_k^{(2)} \end{bmatrix}$, $\bar{H}_{k,k} = \begin{bmatrix} H_{k,k} & H_{k,R_k} \\ H_{k,k} & H_{k,R_k} \end{bmatrix}$ and $\bar{V}_k^{(i)} = \begin{bmatrix} V_k^{(i)} \\ V_{R_k}^{(i)} \end{bmatrix}$

In this way all the channel matrices are now different one from the others, but as in previous solutions, an "intra-matricial" channel correlation is present and the solution can not lead to perfect IA. We need to have different rows to differentiate the output on the two slots. To do this as before, we do not let some device to transmit in some slot, in this way respective unified channel matrices are modified having at least one zero-block and finally perfect IA is reached.[3]

[3] We prefer not to exactly explain the configurations that can lead to a perfect IA but simply let the reader understand how to overcome the problem of correlation in this scenario. The why of this decision will be clear reading throughput analysis.

In both the configurations presented, modifying a little bit the system, perfect IA is reachable, which confirms somehow the results achieved in previous subsection: correlation on channel matrices goes against the possibility to achieve perfect IA.

Throughput analysis

Let us analyze the throughput. In both the configurations presented above it is just of 3.5 pck/slot and less than 5 pck/slot. So it does not seem it is possible to have a throughput increment increasing the number of slots. Is this a general result? We try to generalize starting again from Jafar's inequality to have a more complete description of the overall situation. As told before we understand it is possible to unify transmitters with their relays as unique transmitters, what is new is that the receivers can be unify in the time domain as they would be unique receivers in just one slot. As a consequence the number of transmitting and receiving antennas can be seen respectively as $\bar{M}_t = (N_k + 1) * M_t$ and $\bar{M}_r = slot * M_r$. So Equation (3.7) develops in

$$slot * M_r + (N_k + 1) * M_t - (K + 1)S_k = 0 \tag{3.12}$$

where we used the equal sign to stress the maximum capability achievable by the system. From previous equation we have:

$$(K + 1)S_k = (slot + N_k + 1) * M \tag{3.13}$$

where $M = M_t = M_r$ as usual in our examples is the number of antennas and the first member is the term to be maximized which expresses the number of packets sent in all the slots. To have the throughput from equation above we do a normalization for the number of slots:

$$Thr = \frac{(slot + N_k + 1) * M}{slot} \tag{3.14}$$

This equation states that is not useful to work with configurations exploiting more than one slot. This result seems to be a little bit strange: from information theory we know that decoding in more than one slot is more useful than just in one. Anyway anything wrong is found in the analysis and as a conclusion of this subsection we can say that a real capacity improvement is not found, we can say only to have found cooperative solutions that can reach well-known configurations with more degrees of freedom exploiting particular hypothesis.

Chapter 4

Convergence speed analysis

Another topic of our work is the exploration of how cooperation can be useful to improve the system convergence speed. The iterative algorithms when IA is perfect, i.e. when the degrees of freedom (DoFs) are enough to reach a theoretically infinite SIR, have a convergence speed which depends on the specific analyzed configuration. In many cases when most of the DoFs are saturated, the algorithms could take a really high number of iterations to converge. This is why it could be reasonable to see if equivalent configurations in terms of throughput (number of packets per slot), but exploiting cooperation through relays, could improve this performance.

4.1 Scenario 3

First we analyze Scenario 3, where several *nV-network,* with only one relay associated to each transmission node, are present. Recalling briefly the system model, it could be seen as follows:

$$y_k = \bar{H}_{k,k}\bar{V}_k\widetilde{x}_k + int \tag{4.1}$$

where $\bar{H}_{k,k} = \begin{bmatrix} H_{k,k} & H_{k,R_k} \end{bmatrix}$ and $\bar{V}_k = \begin{bmatrix} V_k \\ V_{R_k} \end{bmatrix}$, and the system could be represented as a *K-network* configuration with a total number of trasmission antennas that is now the number of transmission antennas plus the number of relay antennas. On the following table the results of this approach:

Table 4.1: Analysis on configurations transmitting the same packet at the same time

configuration	*SINR*	*Residual Interf.*	*Failure Rate*	*Fake Rate*	*N*[1]	*Spectral Efficiency*
(3c, 2a, 1sl, 1str_rx) 3V-network	inf	0	0%	0%	2	86
(4c, 3a, 1sl, 1str_rx) 4V-network	inf	0	0%	0%	2	116
(5c, 3a, 1sl, 1str_rx) 5V-network	inf	0	0%	0%	2	146
(6c, 4a, 1sl, 1str_rx) 6V-network	inf	0	0%	0%	2	175
(7c, 4a, 1sl, 1str_rx) 7V-network	inf	0	0%	0%	2	205

What can be noticed is that this system works very well: in just two iterations 80 dB SINR is reached and throughput (1 packet per slot per user) is not changed from critical configurations. The explanation of the results is the same of the one of section 3.2 and it has to be researched on the hypothesis made on the scenario, especially the one of mutual channel knowledge of the sources belonging to the same sub-network: in these conditions more DoFs are achievable allowing the system to see a total amount of transmission

[1] *N is the number of iterations to reach 80 dB SINR*

antennas which is

$$\bar{M}_t = M(N_k + 1) \tag{4.2}$$

where, for configurations of Table 4.1, N_k is always equal to 1.

Aware of these results we tried to verify if, in the case of configurations with relays not associated to each sub-network (for example configurations with only one relay in the whole network), convergence speed could be improved with respect to the base case of the *K-network.* It means that these configurations have only few "*V-networks*", because the total number of relays is less than the number of couples. Jafar's formula is not useful in this case because the network is not symmetric anymore. On Fig. 4.1 the plot of SINR for these configurations in the case of $K = 3$ and $K = 5$ is present, where R in the figure refers to the number of relays.

In Table 4.2 the relative simulative campaign of these configurations is reported, confirming the results presented in Fig. 4.1.

Table 4.2: Convergence speed in Scenario 3 with a variable number of *1V-network*

configuration	*SINR*	*Residual Int*	*failure Rate*	*fake Rate*	*N*[2]	*Spectral Efficiency*
(3c, 2a, 1sl, 1str), K-net	inf	0	0%	0%	105	85
(3c, 2a, 1sl, 1str_rx) 3V-net(1 relay per V)	inf	0	0%	0%	2	87
(3c, 2a, 1sl, 1str_rx) 2V-net(1 relay per V)	inf	0	0%	0%	2.385	87

[2]N is the number of iterations to reach 80 db SINR

(3c, 2a, 1sl, 1str_rx) 1V-net(1 relay per V)	inf	0	0%	0%	16.57	87
(4c, 3a, 1sl, 1str), K-net	inf	0	0%	0%	22	116
(4c, 3a, 1sl, 1str_rx) 4V-net(1 relay per V)	inf	0	0%	0%	2	116
(4c, 3a, 1sl, 1str_rx) 3V-net(1 relay per V)	inf	0	0%	0%	2.312	116
(4c, 3a, 1sl, 1str_rx) 2V-net(1 relay per V)	inf	0	0%	0%	6.46	116
(4c, 3a, 1sl, 1str_rx) 1V-net(1 relay per V)	inf	0	0%	0%	10.37	116
(5c, 3a, 1sl, 1str), K-net	inf	0	0%	0%	500	137
(5c, 3a, 1sl, 1str_rx) 5V-net(1 relay per V)	inf	0	0%	0%	2	146
(5c, 3a, 1sl, 1str_rx) 4V-net(1 relay per V)	inf	0	0%	0%	2.62	145

(5c, 3a, 1sl, 1str_rx) 3V-net(1 relay per V)	inf	0	0%	0%	15.32	145
(5c, 3a, 1sl, 1str_rx) 2V-net(1 relay per V)	inf	0	0%	0%	34.15	145
(5c, 3a, 1sl, 1str_rx) 1V-net(1 relay per V)	inf	0	0%	0%	95	145
(6c, 4a, 1sl, 1str), K-net	inf	0	0%	0%	62	175
(6c, 4a, 1sl, 1str_rx) 6V-net(1 relay per V)	inf	0	0%	0%	2	175
(6c, 4a, 1sl, 1str_rx) 5V-net(1 relay per V)	inf	0	0%	0%	2.34	175
(6c, 4a, 1sl, 1str_rx) 4V-net(1 relay per V)	inf	0	0%	0%	9.39	175
(6c, 4a, 1sl, 1str_rx) 3V-net(1 relay per V)	inf	0	0%	0%	13.36	175

(6c, 4a, 1sl, 1str_rx) 2V-net(1 relay per V)	inf	0	0%	0%	20.28	175
(6c, 4a, 1sl, 1str_rx) 1V-net(1 relay per V)	inf	0	0%	0%	34.15	175
(7c, 4a, 1sl, 1str), K-net	inf	0	0%	0%	1520	150
(7c, 4a, 1sl, 1str_rx) 7V-net(1 relay per V)	inf	0	0%	0%	2	205
(7c, 4a, 1sl, 1str_rx) 6V-net(1 relay per V)	inf	0	0%	0%	2.61	205
(7c, 4a, 1sl, 1str_rx) 5V-net(1 relay per V)	inf	0	0%	0%	14.96	205
(7c, 4a, 1sl, 1str_rx) 4V-net(1 relay per V)	inf	0	0%	0%	24.89	205
(7c, 4a, 1sl, 1str_rx) 3V-net(1 relay per V)	inf	0	0%	0%	43.81	205

(7c, 4a, 1sl, 1str_rx) 2V-net(1 relay per V)	inf	0	0%	0%	90	203
(7c, 4a, 1sl, 1str_rx) 1V-net(1 relay per V)	inf	0	0%	0%	236	202

As the reader infers, performance in these configurations is better than standard *K-networks,* convergence speed is faster also with a few number of relays. With only one relay present in the network a real great improvement is achieved and the throughput in the network remains the same as in the base case (*K-network*).

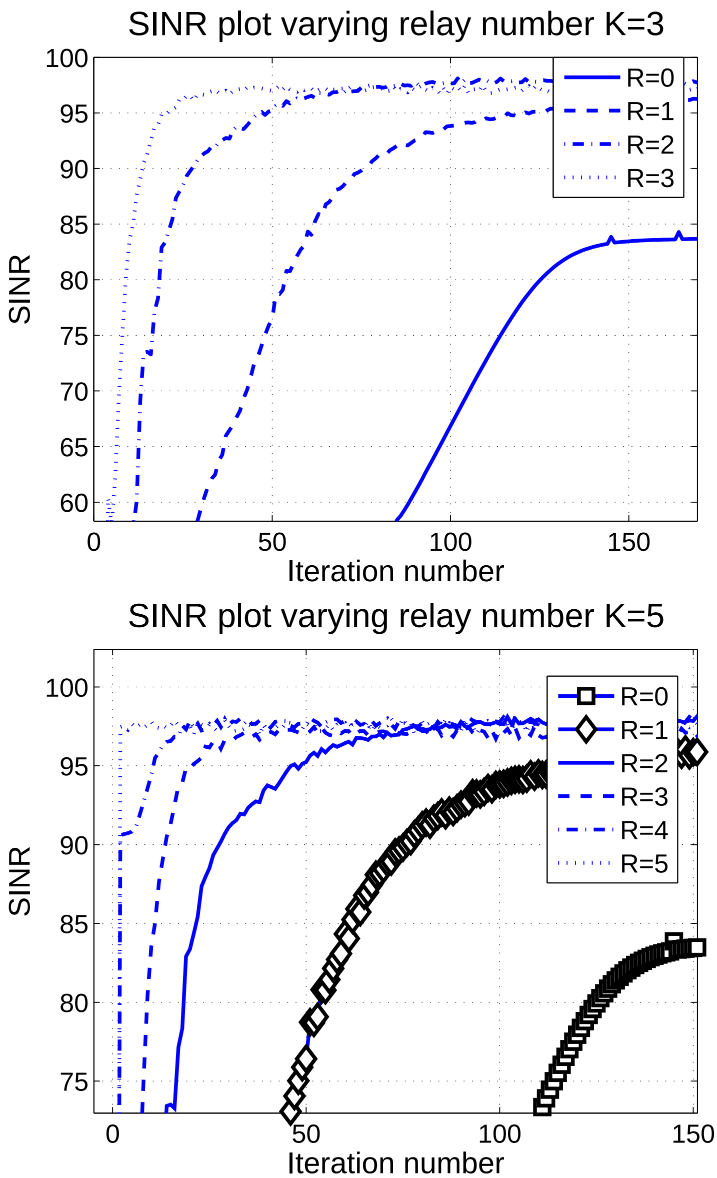

Figure 4.1: Convergence speed in Scenario 3 varing the relay number

4.2 Scenario 2

We move now to the analysis of convergence speed in various Scenario 2 configurations. To do so, as always, we choose some perfect IA *K-network* configurations, trying to understand if, switching to some corresponding *nV-networks* dealing to Scenario 2, an increment of convergence speed is reachable.

4.2.1 Experimental results

At the beginning, another simulative campaign is done, aiming to investigate what stated above. *Zero-Force* and *Max-SINR* algorithms are used and they lead to the same results, while Gradient descend algorithm is too slow and so it is not used for this analysis. The results are reported in Table 4.3 for configurations having $K = \{3, 4, 5, 6, 7, 9, 11\}$.

Table 4.3: Convergence speed analysis on Scenario 2

configuration	*SINR*	*Residual Int*	*Failure Rate*	*fake Rate*	N[3]	*Spectral Efficiency*
(3c, 2a, 1sl, 1str), K-net	inf	0	0%	0%	105	85
(3c, 2a, 2sl, 2str_rx), 3V-net 1 relay per tx-node	inf	0	0%	0%	105	90
(3c, 2a, 3sl, 3str_rx), 3V-net 2 relay per tx-node	inf	0	0%	0%	90	93
(3c, 2a, 4sl, 4str_rx), 3V-net 3 relay per tx-node	inf	0	0%	0%	70	96
(3c, 2a, 5sl, 5str_rx), 3V-net 4 relay per tx-node	inf	0	0%	0%	57	97

[3]N is the number of iterations to reach 80 db SINR

(3c, 2a, 6sl, 6str_rx), 3V-net 5 relay per tx-node	inf	0	0%	0%	47	98
(3c, 2a, 7sl, 7str_rx), 3V-net 6 relay per tx-node	inf	0	0%	0%	44	98
(4c, 3a, 1sl, 1str), K-net	inf	0	0%	0%	22	116
(4c, 3a, 2sl, 2str_rx), 4V-net 1 relay per tx-node	inf	0	0%	0%	18	123
(4c, 3a, 3sl, 3str_rx), 4V-net 2 relay per tx-node	inf	0	0%	0%	14	127
(4c, 3a, 4sl, 4str_rx), 4V-net 3 relay per tx-node	inf	0	0%	0%	13	129
(4c, 3a, 5sl, 5str_rx), 4V-net 4 relay per tx-node	inf	0	0%	0%	11	131

(5c, 3a, 1sl, 1str), K-net	inf	0	0%	0%	500	137
(5c, 3a, 2sl, 2str_rx), 5V-net 1 relay per tx-node	inf	0	0%	0%	324	152
(5c, 3a, 3sl, 3str_rx), 5V-net 2 relay per tx-node	inf	0	0%	0%	216	158
(5c, 3a, 4sl, 4str_rx), 5V-net 3 relay per tx-node	inf	0	0%	0%	171	160
(5c, 3a, 5sl, 5str_rx), 5V-net 4 relay per tx-node	inf	0	0%	0%	150	162
(5c, 3a, 6sl, 6str_rx), 6V-net 5 relay per tx-node	inf	0	0%	0%	136	164
(5c, 3a, 7sl, 7str_rx), 6V-net 6 relay per tx-node	inf	0	0%	0%	127	165

(6c, 4a, 1sl, 1str), K-net	inf	0	0%	0%	62	175
(6c, 4a, 2sl, 2str_rx), 6V-net 1 relay per tx-node	inf	0	0%	0%	47	187
(6c, 4a, 3sl, 3str_rx), 6V-net 2 relay per tx-node	inf	0	0%	0%	40	192
(6c, 4a, 4sl, 4str_rx), 6V-net 3 relay per tx-node	inf	0	0%	0%	37	195
(6c, 4a, 5sl, 5str_rx), 6V-net 4 relay per tx-node	inf	0	0%	0%	35	197
(7c, 4a, 1sl, 1str), K-net	inf	0	0%	0%	1520	150
(7c, 4a, 2sl, 2str_rx), 7V-net 1 relay per tx-node	inf	0	0%	0%	530	209

(7c, 4a, 3sl, 3str_rx), 7V-net 2 relay per tx-node	inf	0	0%	0%	358	221
(7c, 4a, 4sl, 4str_rx), 7V-net 3 relay per tx-node	inf	0	0%	0%	291	225
(7c, 4a, 5sl, 5str_rx), 7V-net 4 relay per tx-node	inf	0	0%	0%	257	228
(7c, 4a, 6sl, 6str_rx), 7V-net 5 relay per tx-node	inf	0	0%	0%	237	230
(7c, 4a, 7sl, 7str_rx), 7V-net 6 relay per tx-node	inf	0	0%	0%	223	232
(9c, 5a, 1sl, 1str), K-net	inf	0	0%	0%	3300	250
(9c, 5a, 2sl, 2str_rx), 9V-net 1 relay per tx-node	inf	0	0%	0%	773	280

(9c, 5a, 3sl, 3str_rx), 9V-net 2 relay per tx-node	inf	0	0%	0%	513	287
(9c, 5a, 4sl, 4str_rx), 9V-net 3 relay per tx-node	inf	0	0%	0%	425	291
(9c, 5a, 5sl, 5str_rx), 9V-net 4 relay per tx-node	inf	0	0%	0%	379	295
(9c, 5a, 6sl, 6str_rx), 9V-net 5 relay per tx-node	inf	0	0%	0%	349.5	298
(11c, 6a, 1sl, 1str), K-net	inf	0	0%	0%	5750	280
(11c, 6a, 2sl, 2str),11V-net 1 relay per tx-node	inf	0	0%	0%	1059	344
(11c, 6a, 3sl, 3str), 11V-net 2 relay per tx-node	inf	0	0%	0%	693	352

(11c, 6a, 4sl, 4str), 11V-net 3 relay per tx-node	inf	0	0%	0%	571.5	358
(11c, 6a, 5sl, 5str), 11V-net 4 relay per tx-node	inf	0	0%	0%	513	362

Looking at Table 4.3 it can be inferred that cooperation is useful to improve the convergence speed: by adding relays in the network (passing from a *K-network* to a *nV-network*) the speed improves (the number N to reach convergence decreases incrementing relays). Cooperation helps convergence speed in every configuration but in different ways depending on the configuration analyzed. We decide to focus only on critical configurations (the ones with an odd number of sub-networks) because they are the most important, having the best throughput with minimum hardware, but they take a high number of iterations, as all the DoFs are saturated. To let the reader have a more visual view of the evolution of convergence speed through configurations, Fig. 4.2 is presented.

Looking at Fig. 4.2, starting from each *K-network* configuration and after adding a certain number of relays, it could seem that the improvement tends to zero, i.e. that number of iterations needed saturates to a value greater than zero. This is not true because simulations with a high number of relays confirm that convergence speed slowly increases also for high N_k. Obviously a good result is achieved when just by adding one relay to each sub-network, a great improvement is reached (for example a reduction of 50% in the number of iterations needed). We remark that the larger the network more this particular feature seems to be achieved: for instance we obtain that for a 7 couples IA-perfect configuration, the reduction adding just a relay per node is over 65% compared with the standard 7 couples *K-network* configuration, while, choosing the 3 couples network in the same conditions, there is no reduction at all.

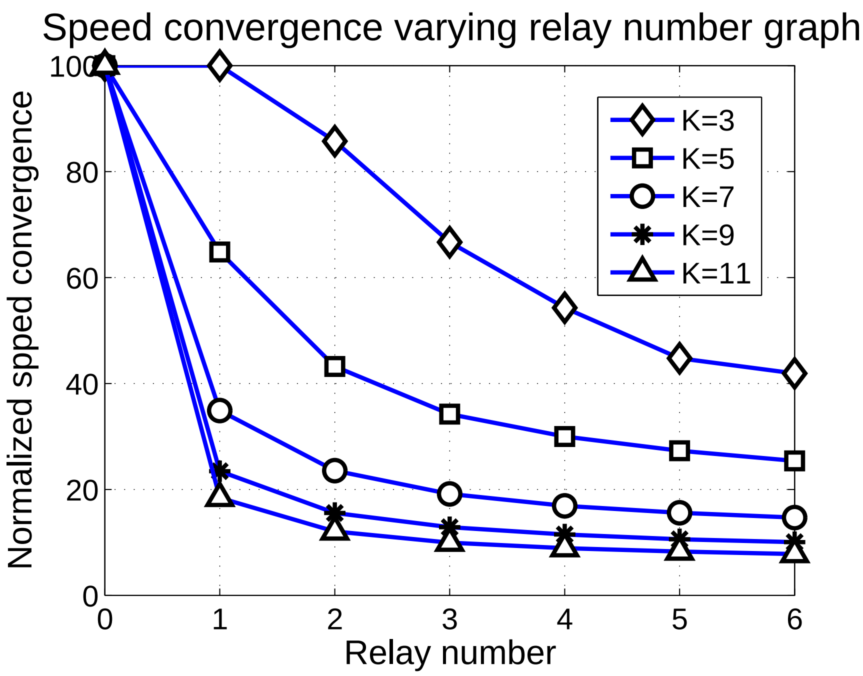

Figure 4.2: Convergence speed varying relay number on Scenario 2

4.2.2 Theoretical explanation

Model and approximations

While in the last section an explanation about the improvement of convergence speed was found, the reason is not clear in this scenario. As a consequence we decide to investigate it.

As both algorithms used lead to the same results, we examine deeper the Zero-Forcing algorithm. Both lemmas, that are exploited to perform the double minimization in Zero-Forcing algorithm, can be interpreted as a gradient algorithm as presented in [16]. Globally, we think that this algorithm works as a steepest descend on which at each iteration the perfect step α is found performing a line search: the α chosen is that one that minimizes the object function along a line as written in [17].

As a consequence we can say that our algorithm works in the following way:

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n) \tag{4.3}$$

If the object function is a quadratic function, that is

$$f(x) = \frac{1}{2} x^T A x - b^T x + c \tag{4.4}$$

where x is a real vector, it is demonstrated in [17] that the norm of the error to reach the solution is inversely exponential with the number of iterations:

$$\frac{||e_{it}||}{||e_0||} = w^{it} \tag{4.5}$$

The error can be expressed as $e_{it} = x_{it} - \hat{x}$, where $\hat{x}$ is the final solution and x_{it} is the solution at iteration it.

w is the base of the exponential. w is always less than 1 and the smaller this parameter, the faster the convergence. As written [17] w depends on the matrix A: as A is better conditioned (cfr. Appendix) as the parameter w will be smaller.

In our case the object function is obviously the residual interference:

$$
\begin{aligned}
INT_{res} = & \sum_{k=1}^{K} \sum_{l \neq k}^{K} ||H_{k,l}V_l - C_k C_k^H H_{k,l} V_l||^2 + \\
& + \sum_{k=1}^{K} \sum_{l \neq k}^{K} \sum_{f=1}^{N_l} ||H_{k,R_l^{(f)}} V_{R_l^{(f)}} - C_k C_k^H H_{k,R_l^{(f)}} V_{R_l^{(f)}}||^2
\end{aligned}
\tag{4.6}
$$

and can also be written as:

$$
\begin{aligned}
INT_{res} \quad = \quad & \mathrm{tr} \sum_{k=1}^{K} \sum_{l \neq k}^{K} [\big(V_l^H H_{k,l}^H U_k U_k^H H_{k,l} V_l\big) + \\
& \sum_{f=1}^{N_l} \Big(V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H U_k U_k^H H_{k,R_l^{(f)}} F_1^H{}_{R_l^{(f)}}\Big)]
\end{aligned}
\tag{4.7}
$$

From the formula above it is easy to find that

$$
INT_{res} = \mathrm{tr}\left(x^H P x\right) \tag{4.8}
$$

where the complex vector x is

$$
x = \begin{bmatrix} V_1 \\ \vdots \\ V_K \\ V_{R_1^{(1)}} \\ \vdots \\ V_{R_K^{(N_k)}} \\ U_1 \\ \vdots \\ U_K \end{bmatrix} \tag{4.9}
$$

P is a block diagonal matrix on which in the first K blocks, depending on transmitters, there are the following matrices:

$$\sum_{k \neq l} H_{k,l}^H U_k U_k^H H_{k,l} \tag{4.10}$$

in the subsequent $K * N_k$ blocks there are the matrices depending on the relays:

$$\sum_{k \neq l} H_{k,R_l^{(f)}}^H U_k U_k^H H_{k,R_l^{(f)}} \tag{4.11}$$

finally in the last K blocks the matrices depending on the receivers are present:

$$\sum_{l \neq k} \left(H_{k,l} V_l V_l^H H_{k,l}^H + \sum_{f=1}^{N_k} H_{k,R_l^{(f)}} V_{R_l^{(f)}} V_{R_l^{(f)}}^H H_{k,R_l^{(f)}}^H \right) \tag{4.12}$$

In this analysis three important things have to be noticed:

- Our object function in Eq. (4.8) is not quadratic, but it is the trace of a simplified quadratic function: it is not a great problem because the analysis of the error behavior in [17] depends more on the gradient of the function than on the function itself and in our case (cfr. Appendix):

$$\nabla_x \left(\mathrm{tr} \left(x^H P x \right) \right) = \nabla_x \left(x^H P x \right) \tag{4.13}$$

- P depends on x, but an approximation can be done saying that P is fixed starting from a certain point on. This is true because after the first iterations IA becomes effective and P is very slowly variant (the gradient becomes smaller going on with iterations)

- It is written above that our complex vector x is the concatenation of all the precoders and decoders. This can be done only in the case of no relays present in the network (the basic case of a *K-network*), because in Scenario 2 the number of streams (and so the number of columns) of decoders is different from the one of precoders (which is always equal to 1 for every configuration on this scenario). The problem can be solved imagining to have two different gradient algorithms: the first optimizes precoders and the second optimizes decoders with two different object

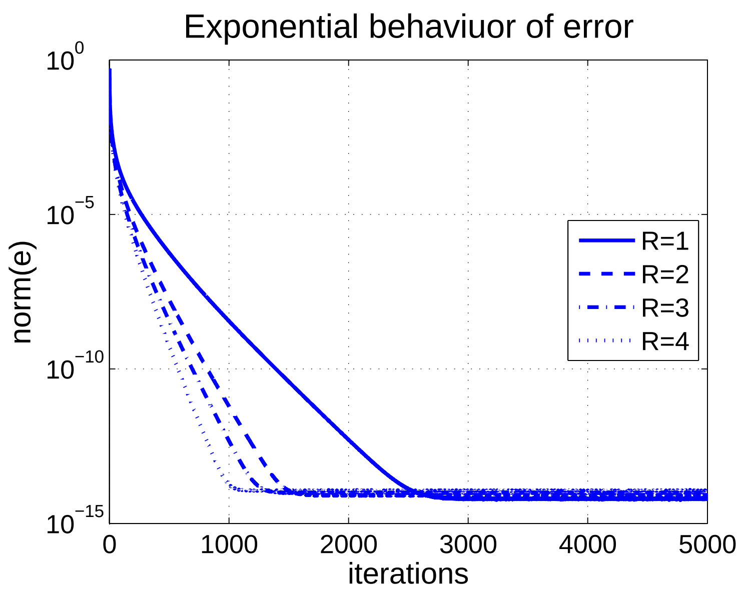

Figure 4.3: Inverse exponential behavior of error for K=7 configuration varing the relay number R

functions, having both the form of Eq. (4.8). Despite P, in the first algorithm there would be P' formed by the first $K(N_k+1)$ columns and rows of P and in the second P" formed by the last K columns and rows of P. With an abuse of notations we will continue using the unified model for simplicity and because only the optimization of the precoders will have consequences on the analysis as it will be explained on the following lines.

P matrix behavior

Starting from this model, does the error to reach the solution have an effective exponential behavior in our case?

From Fig. 4.3, it is possible to see that a similar behavior is confirmed. In the first iterations, reader can observe there is a transient phase on which the error decreases quickly without a comprehensible behavior. This depends on the nature of our matrix P that contains the variables. At the beginning P can be assumed to be random and so full-rank and that is why the minimization is faster. In the first iterations P varies but when the IA procedure computes a first order approximation of the beamformers, that is when the eigenvalues of the matrix P approximate quite well their final values, the exponential behavior is obtained. Finally a saturation zone is present which is simply due to the numerical noise of Matlab. We remark that, as written in Table 4.2, it is in the first iterations that IA is really performed, only after this step, the exponential behavior of the error is obtained.

It is observed in Fig. 4.3 that effectively, adding relays, w gets smaller which confirms that the convergence is faster. As said before going on with IA technique, eigenvalues of P approximate the final ones, but when perfect IA is reachable, as in the cases of interest, P must always have some zero eigenvalues, that means perfect IA always implies a bad-conditioned P matrix. Let us try to better explain this critical step.

Matrix P is always block-diagonal (for the hypotheses made on the scenario) and the eigenvalues of a block-diagonal matrix are the eigenvalues of the single blocks. As written before P is formed by blocks as in Eq. (4.10), (4.11), (4.12).

- Each block of type (4.12) can be interpreted as the interference sub-space of each receiver. Simulation results demonstrate that these blocks have always $N_k + 1$ zero eigenvalues. This is perfectly correct: the dimension of interference sub-space (given by the number of non-vanishing eigenvalues of its matrix) must be lower than the dimension of the overall signal space to ensure the decoding of useful received signals. Since the number of useful signals is exactly $N_k + 1$, equal to the number of streams for each sub-network, the same number of zero-eigenvalues confirms that IA is successful.

- The blocks of type (4.10), (4.11) do not have a physical explanation as the others, but they can be interpreted as the interference matrices imagining an inverse communication, where the sources become the receivers and viceversa. In this way each new receiver (the old source) has only one useful signal to decode and as a consequence this type of matrices has only one zero eigenvalue. The explanation is the same of the one above.

What is really important now is to understand how the rank of the matrix P varies adding relays. To do so, we analyze the ratio between the total number of zero eigenvalues in matrix P and its total dimensions:

$$\frac{\#zeros_P}{dim} = \frac{[K(N_k+1) + K(N_k+1)]}{[K + K\,(N_k+1)]\,M(N_k+1)} = \frac{2}{M(N_k+2)} \tag{4.14}$$

where the first term of the numerator refers to the total number of zero eigenvalues for the blocks of type (4.12), the second one refers to the total number of zero eigenvalues for the blocks of type (4.10) and (4.10), while the first term of the denominator refers to the total number of blocks and the second to the dimensions of each matrix.

From Eq. (4.14) the reader can understand that adding relays the percentage of all zero eigenvalues becomes lower, which implies that adding relays matrix P is “less ill-conditioned” and, as a consequence of the theory explained in the last subsection, it is demonstrated that w decreases.

Asymptotically with the number of relays

$$\lim_{N_k \to \infty} \frac{\#zeros_P}{dim} \to 0 \tag{4.15}$$

the reader can infer that P becomes a full rank matrix.

From Fig. 4.4, 4.5, 4.6, 4.7 the reader can verify that the distribution of zero eigenvalues decreases adding relays: the value of the first bin follows the rule

$$\frac{2}{M(N_k+2)}$$

4.2.3 Further consequences

From this analysis the reader understands that with a *Zero-Forcing* algorithm the norm of the error $||e_{it}||$ follows an inversely exponential behavior after a transient phase. Empirical results demonstrate that also the interference has a similar behavior:

$$I_{it} = I_{t_0} w^{it-t_0} \tag{4.16}$$

as in Fig. 4.8.

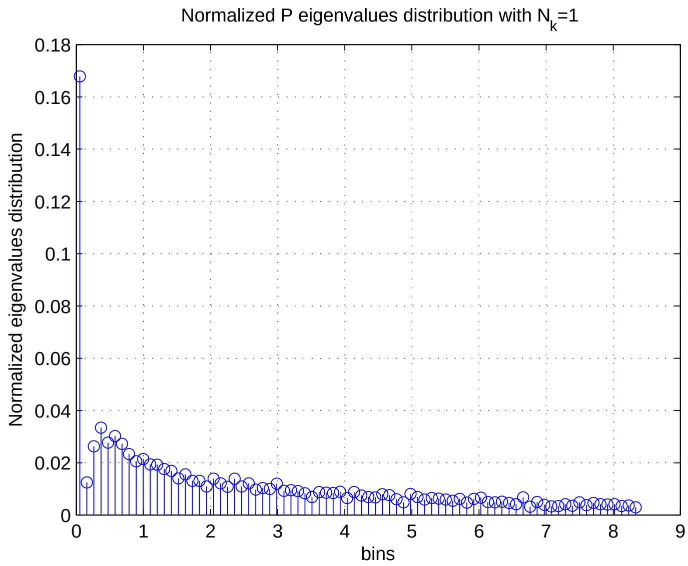

Figure 4.4: Eigenvalues distribution in (7c, 4a, 2sl, 2str_rx)

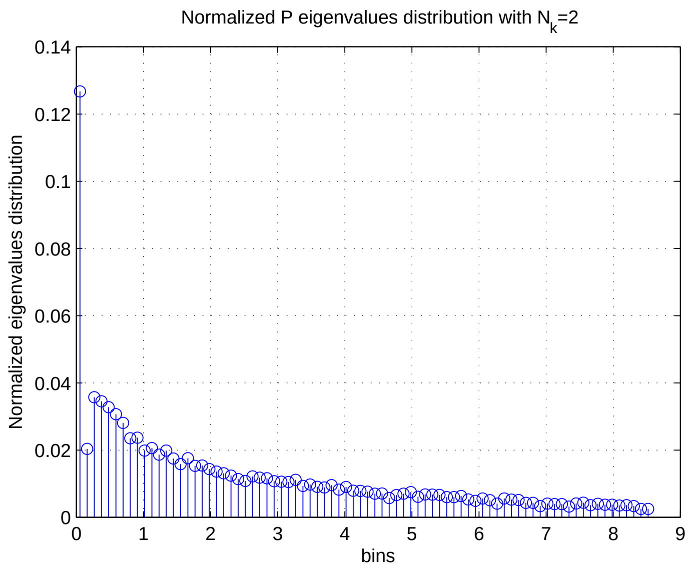

Figure 4.5: Eigenvalues distribution in (7c, 4a, 3sl, 3str_rx)

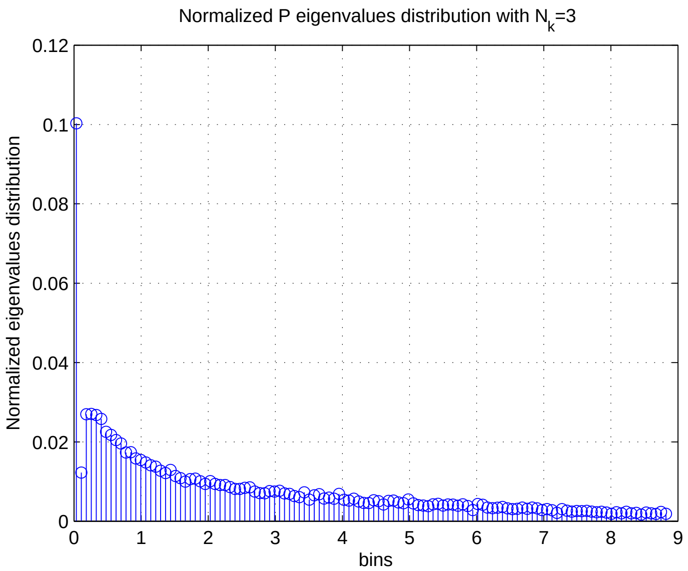

Figure 4.6: Eigenvalues distribution in (7c, 4a, 4sl, 4str_rx)

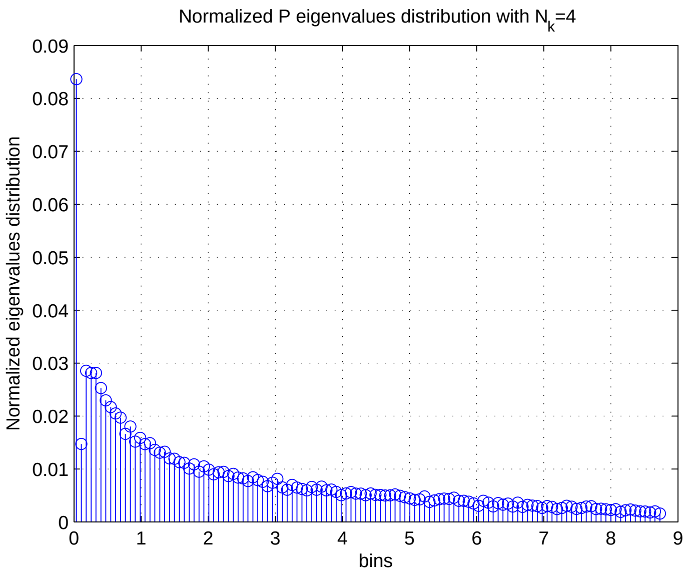

Figure 4.7: Eigenvalues distribution in (7c, 4a, 5sl, 5str_rx)

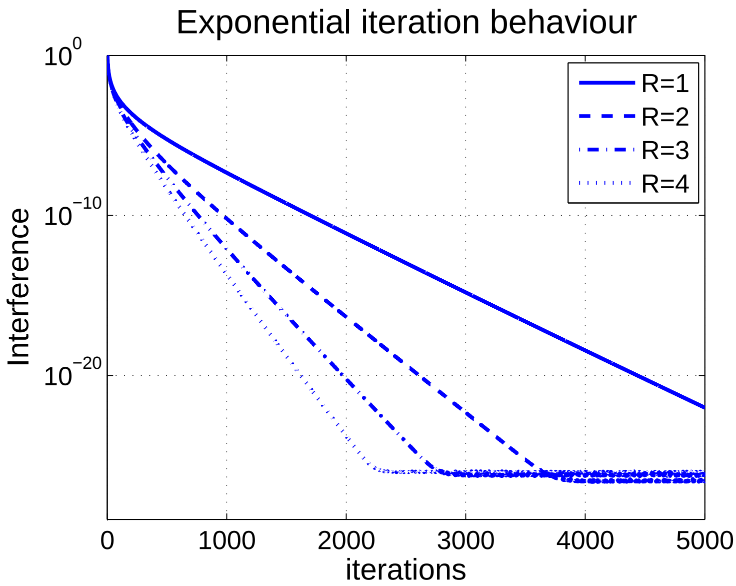

Figure 4.8: Exponential behavior of interference varying the relay number

Notice that the base of the exponential is exactly the same to the one of the error w. The $SINR_{dB}$ can be expressed in the following way:

$$(SINR_{it})_{dB} = (S_{it})_{dB} - N_{dB} - \left(1 + \frac{I_{it}}{N}\right)_{dB} \tag{4.17}$$

As a consequence:

$$(SINR_{it})_{dB} = \begin{cases} 10\log(S_i/N) - 10\log(I_0/N) - (it - t_0)10\log(w) \\ 10\log(S_i/N) - \frac{10}{\ln(10)}\frac{I_0}{N}w^{it-t_0} \end{cases} \tag{4.18}$$

where the first behavior is obtained when $(SINR_{it})_{dB} < 10\log(\frac{1}{N})$ and the other one when $(SINR_{it})_{dB} \geqq 10\log(\frac{1}{N})$. As $10\log(S_i/N)$ can be considered constant, the first behavior is linear, while the second can be viewed as an exponential. Our simulations demonstrate this analytical step and the results are reported in Figs 4.9, 4.10. In this way we can also say to have found several ways to have an empirical estimate of w.

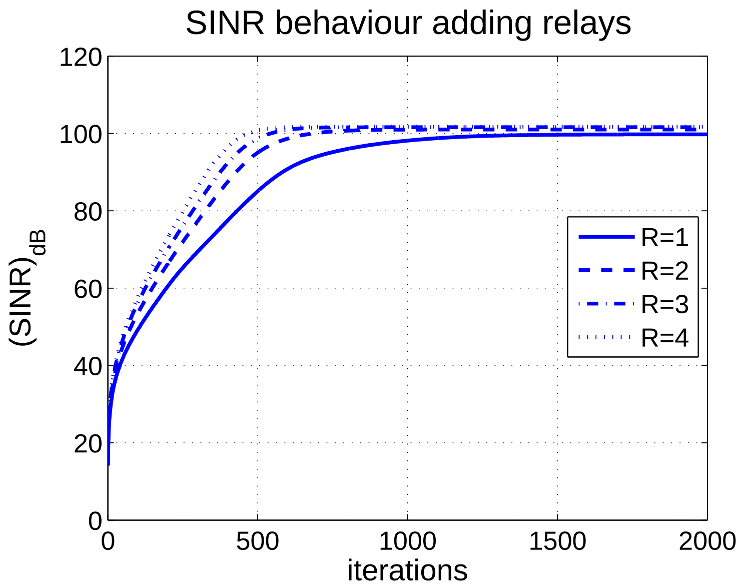

Figure 4.9: Linear behavior of $SINR_{dB}$ in the first iterations varying the relay number

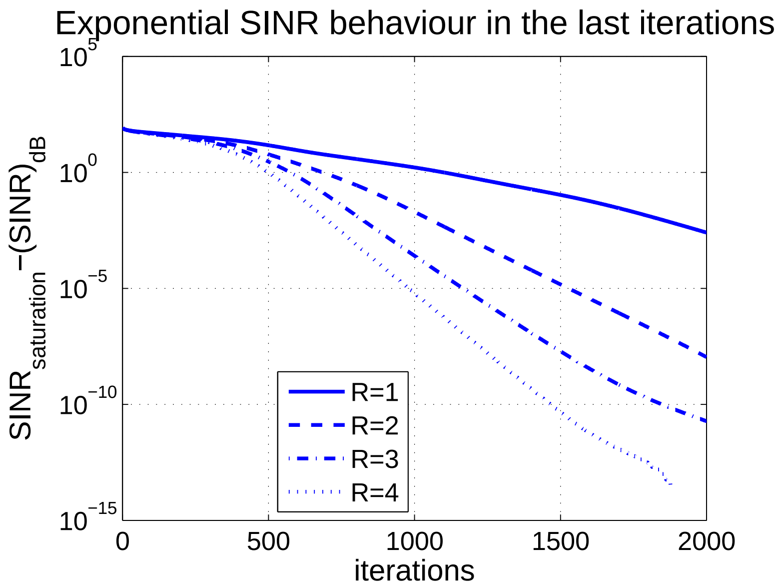

Figure 4.10: Exponential behavior of $SINR_{dB}$ in the last iterations varying the relay number R

4.2.4 Other results

In 4.2.1 and 4.2.2 we did an empirical and a theoretical study of the convergence speed fixing the configuration and adding the relays. Anyway we analyzed the problem from another point of view. Studying with more attention the results we understood that fixing the relay number and let the configuration vary, an approximation similar to both an inverse power and an inverse exponential is obtained. We decided to do a least square fit to understand which of them yields the better approximation.

As it could be inferred by Figs. 4.11, 4.12 and 4.13 the normalized convergence speed decreases as the configuration grows and the inverse power gives a better approximation especially when K is bigger. It is possible that also this behavior can be demonstrated thanks to an analysis similar to the one exploited in the last subsection, but this part is left as future work.

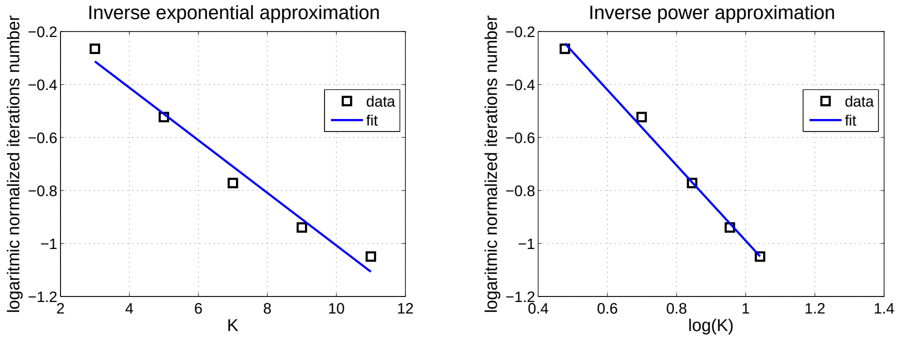

Figure 4.11: Least square approximation with inverse power and inverse exponential with N_k=4 configuration

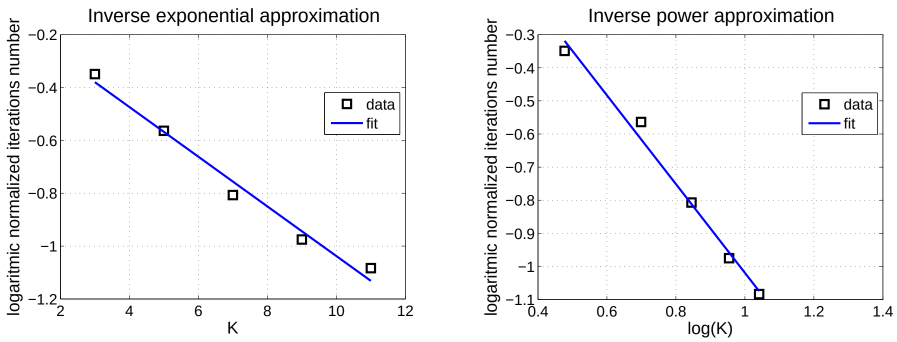

Figure 4.12: Least square approximation with inverse power and inverse exponential with N_k=5 configuration

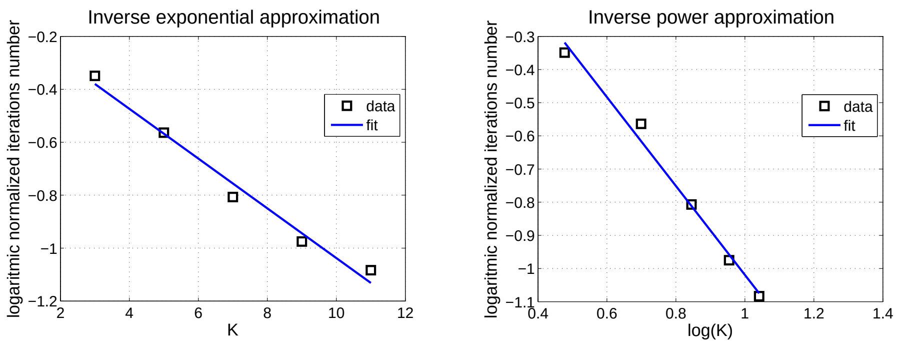

Figure 4.13: Least square approximation with inverse power and inverse exponential with N_k=6 configuration

Chapter 5

Conclusions and future work

In this chapter we summarize our work, we list the results obtained and finally we propose some future works.

5.1 Conclusions

After a study of what is currently present in literature about Interference Alignment, we extended the current model for the cooperative case, adding relays, and in the temporal domain letting the system to work in more than one temporal slot. To do this, we analyzed several cooperative scenarios, each of them made up of some hypothesis on the topology of the network, on the knowledge of sources, on the parameters of the system etc. We studied, generalized or idealized and finally implemented in Matlab some IA algorithms for the cooperative case. Finally we exploited all this background to study both the possibilities of a capacity improvement and an improvement of the convergence speed of algorithms.

Looking first at the capacity improvement, we analyzed several scenarios but due to several reasons the original goal could not be achieved. Nevertheless we understood how correlation of the channel can impact on the performances. Our cooperative scenarios are not able to eliminate it and in scenarios on which correlation was less it was no possible to achieve better results. Anyway we understood also how, exploiting cooperation and hypothesis on mutual channel knowledge of the sources, configurations with more degrees of freedom are achievable and this result is useful for both the capacity improvement and the convergence speed.

The results in terms of convergence speed are more attractive: we discovered that cooperation is truly useful for the improvement of this metric because, adding relays on the sub-networks while keeping the same total throughput, a great improvement on the speed of algorithms is reached. In this case a demonstration through the gradient algorithm properties is provided without exploiting particular hypothesis on channel knowledge. We think this is an interesting result which can shed more light into the properties of IA.

As a consequence of this demonstration, we could deepen the understanding on the behavior of the residual interference and SINR with the *Zero-Forcing* algorithm.

5.2 Future work

In this subsection we suggest some directions for future work, to continue the exploration of IA through cooperation:

- Are there any possibilities to achieve a capacity improvement thanks to cooperation? An analysis of other scenarios could be performed trying to eliminate correlation on channel matrices.

- If no capacity improvement can be achieved in any scenario, a demonstration that cooperation is not useful for this topic could be provided.

- Thanks to the analysis we performed, it is possible to verify if particular locations of the relays can increase the capacity. As we demonstrated empirically, some channel realizations could actually let the system achieve a capacity increase. A future work could be a deeper analysis to understand why some realizations (and as a consequence the relative relay positions) work and other not.

- With respect to our demonstration of convergence speed, does a general law describing how it varies adding relays exist? This is currently under investigation and a first analysis can suggest that a law as

$$\frac{1}{(\log N_k)^{\alpha}} \tag{5.1}$$

could be the right one, but further experimental evidence and theoretical analysis is necessary for more conclusive observations.

- As anticipated in the last chapter, a proof of the empirical relationship that describes the convergence speed against important design parameters could be derived.

Appendix A

Useful mathematical concepts

In the following lines all the mathematical subjects exploited in the thesis will be briefly reminded.

Proprieties of matrices [18]

- A matrix A is said to be hermitian if $A = A^H$.
- A matrix A is said to be normal if $A^H A = AA^H$ and it is constituted of a set of eigenvectors.
- A complex matrix U is said to be unitary if $U^H U = I$.
- An hermitian matrix A is said to be positive definite (pd) if $x^H Ax > 0$ for all non-zero $x \in \mathbb{C}^n$. If $x^H Ax \geq 0$, A is said to be positive semidefinite (psd). If A is pd (spd) all the eigenvalues are positive (nonnegative).

Eigenvalues and eigenvectors [18] If $\lambda \in \mathbb{C}$ and $x \neq 0$ satisfy the equation $Ax = \lambda x$ they are considered eigenvalue and eigenvector of matrix A. The eigenvalue spectrum $\sigma(A)$ of A is the set of all the eigenvalues.

- The eigenvalues of A^H are the complex conjugates of the eigenvalues of A

- The eigenvalues of a diagonal matrix M are the set of all the elements present in the diagonal of M

- The eigenvalues of a block diagonal matrix M are the eigenvalues of each single block of M

Singular value decomposition [18] Every rank-k matrix A may be written as $A = U\Sigma V^H$ where U and V are unitary matrices. The diagonal matrix Σ contains the k non-zero entries $\sigma_1, ..., \sigma_k$ in non-increasing order and zeros elsewhere

- The real singular values σ_1 are the nonnegative square roots of the eigenvalues of AA^H.

- The columns of U are the eigenvectors AA^H, the columns of V are the eigenvectors of $A^H A$.

QR decomposition [18] A rectangular matrix A with dimensions $M \times M - S$ can be written as $A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ where Q_1 $(dim(Q_1) = M \times M - S)$ and Q_2 $(dim(Q_2) = M \times S)$ are both normal and R_1 with dimensions $M - S \times M - S$ is upper triangular matrix. Q_1 constitutes an orthonormal base of A while Q_2 constitutes an orthonormal base of $A^\perp$.

Pseudoinverse [18] The pseudoinverse matrix $A^\dagger$ of A, where A is a rectangular matrix of dimensions $m \times n$, is defined as

$$A^\dagger = \begin{cases} (A^H A)^{-1} A^H & m > n \\ A^H (A^H A)^{-1} & m < n \end{cases}$$

- If $A = U\Sigma V^H \Rightarrow A^\dagger = V\Sigma^\dagger U^H$ where $\Sigma^\dagger$ is Σ^T in which the positive singular values are replaced by their reciprocals.

Trace operator [18] Given a matrix A with elements $a_{i,j}$, dimensions $m \times n$ and eigenvalues λ_i:

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i$$

- $\mathrm{tr}(\alpha A) = \alpha \mathrm{tr}(A)$
- $\mathrm{tr}(A + B) = \mathrm{tr}(A) + \mathrm{tr}(B)$
- $\mathrm{tr}(AB) = \mathrm{tr}(BA)$

Matrix norms [19] Given a matrix A with elements $a_{i,j}$, dimensions $m \times n$, eigenvalues λ_i and single values σ_i, the norm p of A $||A||_p$ with $p = \{1,\, 2,\, \infty,\, F\}$ is defined as follows:

- $||A||_\infty = \max_{||x||_\infty = 1} ||Ax||_\infty = \max_i \sum_j |a_{i,j}|$
- $||A||_1 = \max_{||x||_1 = 1} ||Ax||_1 = \max_j \sum_i |a_{i,j}|$
- $||A||_2 = \sqrt{\rho\left(A^H A\right)}$, $\rho(A) = \max_i |\lambda_i| = \sigma_i$
- $||A||_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{i,j}|^2\right)^{1/2} = \left(\mathrm{tr}\left(A^H A\right)\right)^{1/2} = \left(\sum_{i=1}^{q} \sigma_i^2\right)^{1/2}$ with $q = \min\{m,\, n\}$

Condition numbers [19] The condition number of a matrix A is defined as follows:

$$k(A) = ||A||_p ||A^{-1}||_p \; p = \{1,\, 2,\, \infty,\, F\}$$

If A is singular $k(A) = \infty$

Complex gradient operator [19], [20] Given $z = \begin{bmatrix} z_1 & z_2 & \ldots & z_N \end{bmatrix}^T$ where $z_i = x_i + iy_i$ ($x_i,\, y_i \in \mathbb{R},\, i \in \{1,\, 2,\, \ldots,\, N\}$), the complex operator of the function $f(z)$ is defined as follows

$$\nabla_z f = \begin{bmatrix} \partial f / \partial z_1 & \partial f / \partial z_2 & \ldots & \partial f / \partial z_N \end{bmatrix}^T$$

or

$$\nabla_{z^*} f = \left[\begin{array}{cccc} \partial f/\partial z_1^* & \partial f/\partial z_2^* & \dots & \partial f/\partial z_N^* \end{array} \right]^T$$

where $\partial f/\partial z_i = \frac{1}{2}(\partial f/\partial x_i - \partial f/\partial y_i)$ and $\partial f/\partial z_i^* = \frac{1}{2}(\partial f/\partial x_i + \partial f/\partial y_i)$.

Let $f : \mathbb{C}^N \to \mathbb{R}$ a real-valued function of two complex vector variables. where $f(z) = g(z, z^*)$, where $g : \mathbb{C}^N \times \mathbb{C}^N \to \mathbb{R}$ is a real-valued function of two complex vector variables and g is analytic with respect to each z_k and z_k^*. Then either of the conditions $\nabla_z g = 0$ or $\nabla_{z^*} g = 0$ is necessary and sufficient to determine a stationary point of f. In the following table some main results for the complex gradient are present.

Table A.1: Main results for complex gradient

$\nabla = \nabla_{z^*}$	$\nabla = \nabla_z$
$\nabla(a^H z) = 0$	$\nabla(a^H z) = a^*$
$\nabla(z^H a) = a$	$\nabla(z^H a) = 0$
$\nabla(z^H R z) = Rz$	$\nabla(z^H R z) = R^T z^* = (R^H z)^*$

The results can be extended for the complex gradient of matrices. Let $f(A)$ be a function of the matrix A with elements $a_{i,j}$. The gradient of $f(A)$ is a matrix with elements:

$$\left(\frac{\partial f}{\partial A^*} \right)_{i,j} = \frac{\partial f}{\partial a_{i,j}^*}$$

Some important results we have exploited in this thesis are:

$$\frac{\partial \mathrm{tr}(A^H A)}{\partial A^*} = A$$

$$\frac{\partial \mathrm{tr}(A^H R A)}{\partial A^*} = RA$$

Least square problem [19] Given the vectors x and $\hat{x} = Ac$, the least square problem is to determine $\{c : Ac + e = \hat{x} + e = x\}$. In this way the problem can be written as $\min ||e||_2^2 = \min ||x - Ac||_2^2 \Rightarrow A^H Ac = A^H x$. The optimum c can be found as:

$$c = (A^H A)^{-1} A^H x = A^\dagger x$$

Bibliography

[1] D. Tse and P. Viswanath, “Fundamentals of wireless communications”, *Cambridge Univ. Press*, 2008.

[2] V. Cadambe and S. Jafar, “Interference alignment and degrees of freedom of the K user interference channel”, *IEEE Transactions on Information Theory*, vol. 54, pp. 3425-3441, Aug. 2008.

[3] V. Cadambe and S. Jafar, “Interference alignment and degrees of freedom of wireless X networks”, *IEEE Transactions on Information Theory*, vol. 55, pp. 3893-3908, Sep. 2009

[4] S. Gollakota, S. D. Perli and D. Katabi, “Interference alignment and cancellation”, *ACM SIGCOMM Computer Communication Review*, pp. 159-170, Oct. 2009.

[5] S. W. Peters and R. W. Health, “Interference alignment via alternating minimization”, *IEEE International Conference on Acoustic Speech, and Signal Processing ICASSP*, Apr. 2009.

[6] K. R. Kumar and F. Xue, “An iterative algorithm for joint signal and interference alignment”, *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 2293-2297, Jun. 2010.

[7] K. Gomadam, V. R. Cadambe and S. A. Jafar, “Approaching the capacity of wireless networks through distributed interference alignment,” *Global Telecommunications Conference IEEE GLOBECOM 2008*, pp.1-6, Dec. 2008.

[8] V. Cadambe and S. Jafar, “Can feedback, cooperation, relays and full duplex operation increase the degrees of freedom of wireless networks?”, *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pp. 1263-1267, Jul. 2008.

[9] F. Rossetto and M. Zorzi, "Mixing cooperation and network coding for reliable wireless communications", *IEEE Wireless Communications*, Feb. 2011.

[10] A. Nosratinia, T. E. Hunter and A. Hedayat, “Cooperative communication in wireless networks”, *Communications Magazine, IEEE* pp. 74-81, Oct. 2004.

[11] A. Sendonaris, E. Erkip, and B. Aazhang, “User Cooperation Diversity Part I and Part II,” *IEEE Transaction Communications.*, pp. 1927–48, Nov. 2003.

[12] T. E. Hunter and A. Nosratinia, “Cooperative diversity through coding,” *Proc. IEEE ISIT*, p. 220-228, July 2002.

[13] T. E. Hunter and A. Nosratinia, “Diversity through coded cooperation,” *IEEE Transactions on Wireless Commun*ication, pp. 283-290, Feb. 2006.

[14] C. M. Yetis, T. Gou, S. Jafar and A. Kayran, “Feasibility conditions for interference alignment”, *Proceedings of the Global Communications Conference GLOBECOM*, pp. 1-6, Dec. 2009.

[15] N. Benvenuto, G. Cherubini , “Algorithms for communications systems and their applications”, *John Wiley & Sons*, 2002.

[16] A. Edelman, T. A. Arias, and S. T. Smith, “The geometry of algorithms with orthogonality constraints,” *SIAM Journal on Matrix Analysis and Applications*, pp. 303–353, Oct. 1998.

[17] J. R. Shewchuk, ”An Introduction to the conjugate gradient method without the agonizing pain”, *Technical Report. Carnegie Mellon Univ.,* Aug. 1994.

[18] R. A. Horn, and C. R. Johnson, "Matrix Analysis", *Cambridge University Press*, 1985.

[19] T. Moon and W. Stirling, "Mathematical methods and algorithms for signal processing", *Prentice Hall*, 2000.

[20] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory", *Microwaves Optics and Antennas IEE Proceedings H,* pp. 11-16, 1983.

Table of symbols

K	number of couples transmitter-receiver
M_r	number of receiving antennas
M_t	number of transmitting antennas
l	transmitter index
k	receiver index
j	stream index
$V_l^{(j,s)}$	precoder vector for the l-th transmitter for the j-th stream and for the s-th slot
s	slot index
$U_k^{(j,s)}$	postcoder vector for the k-th receiver for the j-th stream and for the s-th slot
$H_{k,l}$	channel matrix from the l-th transmitter to the k-th receiver
$R_l^{(f)}$	index of the f-th relay associated to the l-th transmitter
f	relay index
n	number of transmitters which have associated one or more relays
S_i	number of stream for the i-th transmitter/receiver
$\widetilde{x}_l^{(j)}$	j-th stream of the l-th transmitter
$\widetilde{y}_k^{(j)}$	j-th stream of the k-th receiver
y_k	k-th receiver received signal
x_l	l-th trasmitter transmitted signal
N_k, R	Number of relays associated at the k-th transmitter

List of Figures

List of Tables