# Distributed Low-Overhead Schemes for Multi-stream MIMO Interference Channels

Hadi Ghauch, *Student Member, IEEE,* Taejoon Kim, *Member, IEEE,* Mats Bengtsson, *Senior Member, IEEE,* and Mikael Skoglund, *Senior Member, IEEE*

***Abstract*—Our aim in this work is to propose fully distributed schemes for transmit and receive filter optimization. The novelty of the proposed schemes is that they only require a few forward-backward iterations, thus causing minimal communication overhead. For that purpose, we relax the well-known leakage minimization problem, and then propose two different filter update structures to solve the resulting non-convex problem: though one leads to conventional full-rank filters, the other results in rank-deficient filters, that we exploit to gradually reduce the transmit and receive filter rank, and greatly speed up the convergence. Furthermore, inspired from the decoding of turbo codes, we propose a turbo-like structure to the algorithms, where a separate inner optimization loop is run at each receiver (in addition to the main forward-backward iteration). In that sense, the introduction of this turbo-like structure converts the communication overhead required by conventional methods to computational overhead at each receiver (a cheap resource), allowing us to achieve the desired performance, under a minimal overhead constraint. Finally, we show through comprehensive simulations that both proposed schemes hugely outperform the relevant benchmarks, especially for large system dimensions.**

***Index Terms*—Distributed algorithms, MIMO Interference Channels, Interference Leakage minimization, Forward-Backward algorithms, Iterative Weight Update, Turbo Optimization**

## I. Introduction

Although the problem of (joint) precoder optimization is an old one, it was not until the recent research on multi-user techniques for multiple-input multiple-output interference channels (MIMO IC), such as Coordinated Multipoint [1] and Interference Alignment (IA) [2], that the problem got mass attention. Since the latter techniques require transmitters and receivers to coordinate their signals, this has given rise to a plethora of centralized or distributed algorithms that attempt to (jointly) optimize the transmit and receive filters, given a predetermined performance metric. Usually, they can be categorized according to the metric that they optimize: such metrics mainly include (weighted) interference leakage [3], [4], (weighted) mean-squared error [5], [6], signal to interference plus noise ratio [3], [7], and (weighted) sum-rate [8], [6] (an insightful and comprehensive comparison of such schemes was done in [9]). Despite the fact that the latter methods attempt to solve a problem that is more generic than Interference Alignment (in a sense that they do not aim at suppressing interference completely), in many of the above cases, there indeed exists an intimate relation between the two: for instance, in the high-SNR sum-rate maximization problem, the precoder optimization problem reduces to finding transmit and receive filters, that satisfy the IA conditions (as formulated in [3]).

However, as the research on IA progressed, it was quickly revealed that many challenges have to be addressed, before any of the promised gains could be harnessed. Such challenges include the need for global channel knowledge at each transmitter, feasibility conditions for the existence of solutions to the IA conditions [10], the absence of closed-form beamforming solutions for generic systems, and whether limited feedback could achieve the optimal degrees-of-freedom promised by IA, [11], [12]. Consequently, people turned their attention to developing distributed schemes that rely on forward-backward (F-B) iterations (e.g., [3], [5]–[8]), since they address most of those challenges. Though the latter works are among the first to use this particular F-B structure within the context of IA, its usage is attributed to many earlier works such as [13], [14]. In brief, each of the so-called F-B iterations exploits the reciprocity of the network - which only holds in systems employing Time-Division Duplexing (TDD), and local Channel State Information (CSI) at each node, to gradually refine each of the transmit and receive filters, one at a time (the receive filters are optimized in the forward training phase, and then transmit filters are optimized in the reverse training phase). Due to the fact that most of those schemes require a relatively large number of such iterations (that seem to increase with the dimensions of the system), this inevitably raises the question of the associated overhead[1]. Despite the plethora of such schemes that implicitly employ this structure, this major issue has not been properly addressed yet.

This issue is the main motivation for the work proposed here: the schemes that are detailed below only require a few F-B iterations, while still delivering large gains in sum-rate

Part of this work has been performed in the framework of the FP7 project ICT-317669 METIS, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues in METIS, although the views expressed are those of the authors and do not necessarily represent the project.

H. Ghauch, M. Skoglund, and M. Bengtsson are with the School of Electrical Engineering and the ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden. E-mails: ghauch@kth.se, mats.bengtsson@ee.kth.se, skoglund@kth.se

T. Kim is with Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong. E-mail: taejokim@cityu.edu.hk

[1]Although many other works consider a more comprehensive definition of overhead (such as [15] and [16] ), we adopt a more simplistic definition of overhead, as the required number of F-B iterations for the algorithm to converge (keeping in mind that the actual overhead will be dominated by this quantity).

performance, w.r.t. the well-known distributed IA algorithm in [3] (the most relevant benchmark). Note that although the authors in [17] used the interference leakage as a metric, their formulation entails a constraint on the desired signal space, rather than having a 'pure' leakage-based scheme such as distributed IA. By parametrizing the leakage at each receiver as a function of some filter parameters (abstracted as $\boldsymbol{A}, \boldsymbol{B}$ in Fig. 1), our proposed schemes can alternately be optimized within the turbo iteration, thereby decreasing the leakage at the corresponding receiver. Furthermore, the exact same structure is used to optimize the transmit filters. Thus, in addition to the F-B iteration used by the above conventional methods, we propose the use of a so-called turbo iteration, where the transmit / receive filters are gradually refined. The introduction of this mechanism greatly speeds up the convergence, and allows us to achieve the desired performance with a strikingly small number of F-B iterations. For that purpose, we propose two different update structures, one resulting in full-rank filters, while the other, possibly, in rank-deficient ones. Although this might seem counter-intuitive at a first glance, we exploit the rank-deficient update structure to "simplify" the alignment, further enhancing the convergence speed. Finally, we compare both algorithms and conclude that although both schemes greatly outperform the benchmark in the low-overhead regime (especially as the dimensions of the problem grow), combining the turbo iteration with the rank-deficient update structure provides the best performance.

In the following, we use bold upper-case letters to denote matrices, and bold lower-case denote vectors. Furthermore, for a given matrix $\boldsymbol{A}$, $[\boldsymbol{A}]_{i:j}$ denotes the matrix formed by taking columns $i$ to $j$, of $\boldsymbol{A}$, $\text{tr}(\boldsymbol{A})$ denotes its trace, $\|\boldsymbol{A}\|_F^2$ its Frobenius norm, $|\boldsymbol{A}|$ its determinant, and $\boldsymbol{A}^\dagger$ its conjugate transpose. In addition, $\lambda_i[\boldsymbol{Q}]$ denotes the $i^{th}$ eigenvalue of a Hermitian matrix $\boldsymbol{Q}$ ( assuming the eigenvalues are sorted in increasing order), and $\mathcal{U}(n,k)$ denotes the set of unitary matrices, i.e. $\mathcal{U}(n,k) = \{\boldsymbol{A} \in \mathbb{C}^{n\times k} \mid \boldsymbol{A}^\dagger\boldsymbol{A} = \boldsymbol{I}_k,\ k \leq n\}$. Finally, $V^\perp$ denotes the orthogonal complement of a subspace $V$, while $card(\mathcal{S})$ denotes the cardinality of a set $\mathcal{S}$.

## II. System Model and Problem Formulation

We consider a $K$-user $M \times N$ MIMO interference channel (IC), where the received signal (after applying the receive filter), is given by

$$\hat{\boldsymbol{x}}^{[k]} = \boldsymbol{U}^{[k]^\dagger}\boldsymbol{H}^{[kk]}\boldsymbol{V}^{[k]}\boldsymbol{x}^{[k]} + \boldsymbol{U}^{[k]^\dagger}\sum_{\substack{j=1,\\ j\neq k}}^{K}\boldsymbol{H}^{[kj]}\boldsymbol{V}^{[j]}\boldsymbol{x}^{[j]} + \boldsymbol{U}^{[k]^\dagger}\boldsymbol{n}^{[k]},$$

where the first term represents the desired signal, and the second one denotes undesired inter-user interference. In the above, $\boldsymbol{H}^{[kj]}$ is the $N \times M$ channel matrix from transmitter $j$ to receiver $k$, $\boldsymbol{V}^{[j]}$ and $\boldsymbol{U}^{[k]}$ are the $M \times d$ and $N \times d$ transmit and receive filters of transmitter $j$ and receiver $k$ ($(k,j) \in \{1,...,K\}^2$), respectively. Furthermore, $\boldsymbol{n}^{[k]}$ is the $N$-dimensional zero-mean AWGN vector with covariance matrix $\sigma^2\boldsymbol{I}_N$, and $\boldsymbol{x}^{[k]}$ the $d$-dimensional vector of transmit symbols intended to receiver $k$, with covariance matrix $E[\boldsymbol{x}^{[k]}\boldsymbol{x}^{[k]^\dagger}] = (\rho/d)\boldsymbol{I}_d$, where $\rho$ is the transmit power, and $\rho/\sigma^2 \triangleq$ SNR. We assume a TDD architecture, where channel reciprocity holds.

### A. Leakage Minimization as a surrogate problem for Sum-Rate Maximization

With the above in mind, the achievable rate of communication for each user is given by,

$$R^{[k]} = \log_2\left|\boldsymbol{I}_d + \left(\boldsymbol{U}^{[k]^\dagger}\boldsymbol{R}_s^{[k]}\boldsymbol{U}^{[k]}\right)\left(\boldsymbol{U}^{[k]^\dagger}(\boldsymbol{Q}^{[k]} + \sigma^2\boldsymbol{I}_N)\boldsymbol{U}^{[k]}\right)^{-1}\right|,$$

where $\boldsymbol{R}_s^{[k]} = (\rho/d)\boldsymbol{H}^{[kk]}\boldsymbol{V}^{[k]}\boldsymbol{V}^{[k]^\dagger}\boldsymbol{H}^{[kk]^\dagger}$ and $\boldsymbol{Q}^{[k]} = \sum_{j\neq k}(\rho/d)\boldsymbol{H}^{[kj]}\boldsymbol{V}^{[j]}\boldsymbol{V}^{[j]^\dagger}\boldsymbol{H}^{[kj]^\dagger}$ are the signal and interference covariance matrix at receiver $k$[2]. As $\sigma^2 \to 0$ (high-SNR regime), the achievable rate $R^{[k]}$ can be approximated by,

$$\tilde{R}^{[k]} = \log_2\left|\boldsymbol{U}^{[k]^\dagger}\boldsymbol{R}_s^{[k]}\boldsymbol{U}^{[k]}\right| - \log_2\left|\boldsymbol{U}^{[k]^\dagger}\boldsymbol{Q}^{[k]}\boldsymbol{U}^{[k]}\right|$$

Then, one can formulate the high-SNR sum-rate maximization problem as follows,

$$(SRM) \quad \max_{\{\boldsymbol{U}^{[k]}\},\{\boldsymbol{V}^{[k]}\}} \quad \tilde{R}_\Sigma = \sum_{k=1}^{K}\tilde{R}^{[k]}. \tag{2}$$

Note that in this work, we only focus on optimizing the interference subspace (as previously proposed algorithms in [3], [7]). Thus, by dropping the signal term in $\tilde{R}^{[k]}$, we can bound it as follows,

$$\tilde{R}^{[k]} \geq -\log_2\left|\boldsymbol{U}^{[k]^\dagger}\boldsymbol{Q}^{[k]}\boldsymbol{U}^{[k]}\right| \overset{(a)}{\geq} \sum_{i=1}^{d} -\log_2\left([\boldsymbol{U}^{[k]^\dagger}\boldsymbol{Q}^{[k]}\boldsymbol{U}^{[k]}]_{ii}\right)$$

$$\overset{(b)}{>} -\sum_{i=1}^{d}[\boldsymbol{U}^{[k]^\dagger}\boldsymbol{Q}^{[k]}\boldsymbol{U}^{[k]}]_{ii} = -\text{tr}(\boldsymbol{U}^{[k]^\dagger}\boldsymbol{Q}^{[k]}\boldsymbol{U}^{[k]})$$

where $(a)$ follows directly from applying Hadamard's inequality, i.e. $|\boldsymbol{A}| \leq \prod \boldsymbol{A}_{ii}$ for $\boldsymbol{A} \succeq \boldsymbol{0}$, and $(b)$ from the fact that $x > \log_2(x),\ \forall\ x > 0$. Although this result is expected, it proves that *minimizing the interference leakage at each user, results in optimizing a lower bound on the user's high-SNR rate*.

### B. Problem Formulation

Now that we have motivated the leakage minimization problem, we turn out attention to devising an iterative algorithm for that purpose. As mentioned earlier, the schemes that we study in this work, fall under the category of distributed schemes, where each receiver / transmitter optimizes its filter, based on the estimated interference covariance matrix. In other words, at the $l^{th}$ F-B iteration, after estimating and updating its interference covariance matrix, $\boldsymbol{Q}_l^{[k]} \leftarrow \boldsymbol{Q}_{l+1}^{[k]}$, receiver $k$ aims to update its filter, $\boldsymbol{U}_l^{[k]} \leftarrow \boldsymbol{U}_{l+1}^{[k]}$, such as to optimize some predetermined metric (interference leakage, mean-squared error, sum-rate, etc...). The F-B iteration structure was first applied within the context of IA, in the distributed IA algorithm

[2]Similarly, we define the interference covariance matrix at transmitter $k$, as follows,

$$\bar{\boldsymbol{Q}}^{[k]} = \sum_{j\neq k}\boldsymbol{H}^{[jk]^\dagger}\boldsymbol{U}^{[j]}\boldsymbol{U}^{[j]^\dagger}\boldsymbol{H}^{[jk]},\ \ \forall\ k = 1,...,K\ . \tag{1}$$

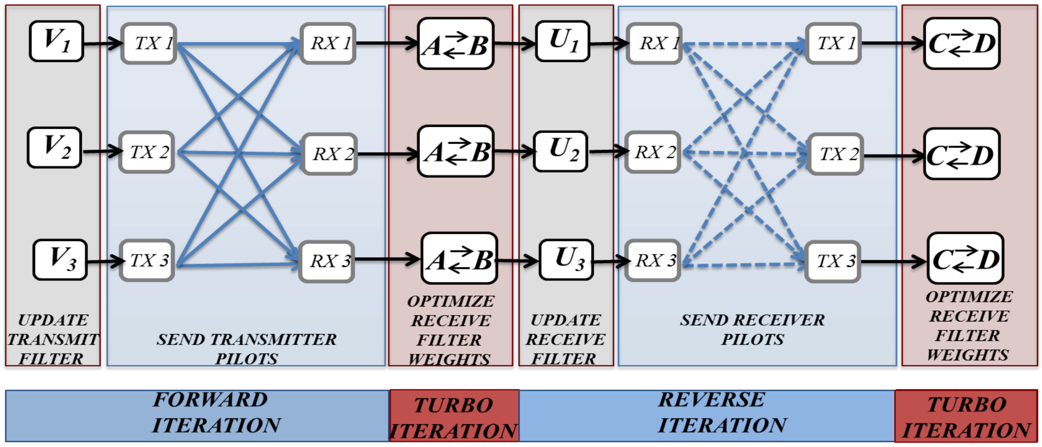


Fig. 1: Proposed Algorithm Structure

(proposed in [3] and re-written below for later reference), where each receive filter update is such that,

$$\min_{\boldsymbol{U}_{l+1}^{[k]}} f^{[k]}(\boldsymbol{U}_{l+1}^{[k]}) = \text{tr}(\boldsymbol{U}_{l+1}^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{U}_{l+1}^{[k]})$$
$$\text{s.t. } \boldsymbol{U}_{l+1}^{[k]\dagger}\boldsymbol{U}_{l+1}^{[k]} = (P_r/d)\boldsymbol{I}_d \ , \qquad (3)$$

where $P_r$ is the receive filter power constraint. In other words, in the forward phase each receiver estimates its interference covariance matrix and updates its filter such as to minimize the interference leakage. Then, in the backward phase, exploiting channel reciprocity, transmitters estimate their respective interference covariance matrices, and use the same update rule of minimizing the leakage. It can be shown that this iteration process, will converge to stationary points of the leakage function. Thus, for the interference leakage cost function, F-B iterations can be used to gradually refine the transmit and receive filters, thereby ultimately creating a $d$-dimensional interference-free subspace at every receiver. Ideally, as $l \to \infty$, the transmit and receive filters that the algorithm yields should satisfy the following IA conditions [2],

$$\text{rank}(\boldsymbol{U}_l^{[k]\dagger}\boldsymbol{H}^{[kk]}\boldsymbol{V}_l^{[k]}) = d \ , \ \forall k = 1, ..., K,$$
$$\boldsymbol{U}_l^{[k]\dagger}\boldsymbol{H}^{[kj]}\boldsymbol{V}_l^{[j]} = \boldsymbol{0} \ , \ \forall j \neq k$$

The existence of transmit and receive filters that fulfil this condition is guaranteed, if the system is feasible (as described in [10]). The distributed IA algorithm has been extensively used and experimentally observed to closely match the theoretical predictions of IA, in small to moderate network configurations. However, one can see that as the dimensions of the problem grow (more antennas and streams), better performance can be achieved be relaxing the unitary constraint.

This sub-optimal performance in multi-stream settings, is partly attributed to the fact that all the streams are allocated the same power - an inherent property of the unitary constraint in (3). It is evident at this point that much could be gained from allocating different powers to different streams, especially as the number of such streams grow, i.e., as $d$ increases. Consequently, we propose to relax the unitary constraint in (3), and allow the transmit / receive filter columns to have unequal norms, i.e., the receive filter update $\boldsymbol{U}_l^{[k]} \leftarrow \boldsymbol{U}_{l+1}^{[k]}$, is as follows,

$$\min_{\boldsymbol{U}_{l+1}^{[k]}} f^{[k]}(\boldsymbol{U}_{l+1}^{[k]}) = \text{tr}(\boldsymbol{U}_{l+1}^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{U}_{l+1}^{[k]})$$
$$\text{s.t. } \|\boldsymbol{U}_{l+1}^{[k]}\|_F^2 = P_r. \qquad (4)$$

Note that the factor $(P_r/d)$ in (3) ensures that the receive power constraint, $\|\boldsymbol{U}_{l+1}^{[k]}\|_F^2$, is the same for both (3) and (4). Let $\mathcal{R}$ and $\mathcal{S}$ be the feasible sets of (3) and (4) respectively, i.e., $\mathcal{R} = \{\boldsymbol{U} \in \mathbb{C}^{N\times d} \mid \boldsymbol{U}^\dagger\boldsymbol{U} = (P_r/d)\boldsymbol{I}_d\}$ and $\mathcal{S} = \{\boldsymbol{U} \in \mathbb{C}^{N\times d} \mid \text{tr}(\boldsymbol{U}^\dagger\boldsymbol{U}) = P_r\}$. Consequently, for any $\boldsymbol{U} \in \mathcal{R} \Rightarrow \boldsymbol{U}^\dagger\boldsymbol{U} = (P_r/d)\boldsymbol{I}_d \Rightarrow \text{tr}(\boldsymbol{U}^\dagger\boldsymbol{U}) = P_r \Rightarrow \boldsymbol{U} \in \mathcal{S}$. This implies that $\mathcal{R} \subseteq \mathcal{S}$, and that indeed (4) is a relaxation of (3). In addition, note that the distributed IA problem in (3) has a simple analytical (well-known) solution. Although the reformulation in (4) promises to deliver better performance, it does make the problem non-convex.

In spite of this non-convexity, the problem can still be tackled in many ways. Firstly, note that (4) can in fact easily be solved by writing the problem in vector form and finding the globally optimal rank-one solution spanned by the eigenvector of $\boldsymbol{Q}_{l+1}^{[k]}$ with the minimum eigenvalue. In addition, it is also known that in the case of (4), Semi-Definite Relaxation (SDR) provides the optimal solution as well [18]. However, the solution that both these methods yield is rank-one[3], and it is well-known from the interference alignment literature that the optimal filter rank in the high-SNR regime is $d$ (assuming that $d$ has been selected properly such that the system is feasible). On the other hand, at medium and low-SNR, the sum-rate performance will improve if the filters have reduced rank (in the limit, the waterfilling power allocation results in one stream being active, in the very low-SNR). The main idea behind our proposed algorithm is therefore to *not* solve (4) but rather to use it as a heuristic, while preventing the algorithm from always converging to

[3]Since the rank is a coarse measure, we use a wider definition of the rank of a matrix, throughout this paper. Let $\boldsymbol{A} \in \mathbb{C}^{n\times m}(n > m)$, then we define $rank(\boldsymbol{A}) = card(\{\sigma_i(\boldsymbol{A}) \mid \sigma_i(\boldsymbol{A}) > \delta, \ \forall\ i = 1, ..., m\})$, where $\{\sigma_1(\boldsymbol{A}), ..., \sigma_m(\boldsymbol{A})\}$ are the singular values of $\boldsymbol{A}$, and $\delta$ a predetermined tolerance.

the aforementioned rank-one solution of (4), either explicitly using a rank-preserving algorithm or implicitly by exploiting the transient phase of the rank-reducing algorithm and stopping after a small number of iterations (more about this in Sect. III). As a result, those algorithms should give a better performance than the optimal solution to (4) given above (simulations will show that this claim is indeed true).

Thus, imposing two different update rules on the transmit / receive filters yields the two different algorithms mentioned above: while one of the update rules do not necessarily result in full-rank transmit / receive filters (which we refer to as *rank-reducing updates*), the other one implicitly enforces full-rank transmit / receive filters (which we refer to as *rank-preserving updates*). The reason for this distinction, as well as its impact, will become clearer in Sect. IV-A.

## III. Proposed Scheme for rank-reducing Updates

Within this class, we opted to use the most generic update rule (i.e., the one that represents the "widest" class of matrices), for obvious reasons. Thus, we propose the following update structure,

$$\boldsymbol{U}_{l+1}^{[k]} = \boldsymbol{\Delta}^{[k]}\boldsymbol{A}_l^{[k]} + \boldsymbol{\Phi}^{[k]}\boldsymbol{B}_l^{[k]}, \tag{5}$$

where $\boldsymbol{\Delta}^{[k]} \in \mathcal{U}(N,d)$ and $\boldsymbol{\Phi}^{[k]} \in \mathcal{U}(N,N-d)$ are such that $\boldsymbol{\Delta}^{[k]\dagger}\boldsymbol{\Phi}^{[k]} = \mathbf{0}$. Furthermore, $\boldsymbol{A}_l^{[k]} \in \mathbb{C}^{d\times d}$ and $\boldsymbol{B}_l^{[k]} \in \mathbb{C}^{(N-d)\times d}$ are the combining weights of $\boldsymbol{\Delta}^{[k]}$ and $\boldsymbol{\Phi}^{[k]}$, respectively.[4] We underline the fact that some choices of $\boldsymbol{\Delta}^{[k]}$ and $\boldsymbol{\Phi}^{[k]}$ should be better than others, in terms of cost function value. Although this would suggest that they should be optimized within each iteration, a quick look at the resulting optimization problem reveals that the complexity of such a scheme would be tremendously high. As a result, we opt to have the sets $\{\boldsymbol{\Delta}^{[k]}\}$ and $\{\boldsymbol{\Phi}^{[k]}\}$ fixed throughout the algorithm. In addition to the fact that the update rule in (5) is the most generic possible (i.e., it can represent any matrix), another reason for picking such a structure is that the resulting optimization problem is a relaxation (although a non-convex one) of the optimization solved by the distributed IA [3] - a result that is formalized in the next subsection.

### A. Relaxation Heuristic

By incorporating the update in (5) into (4), the resulting optimization problem is given by,

$$\begin{aligned} \min_{\boldsymbol{U}_{l+1}^{[k]}} \; & f^{[k]}(\boldsymbol{U}_{l+1}^{[k]}) = \mathrm{tr}(\boldsymbol{U}_{l+1}^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{U}_{l+1}^{[k]}) \\ \text{s.t.} \; & \|\boldsymbol{U}_{l+1}^{[k]}\|_F^2 = P_r \\ & \boldsymbol{U}_{l+1}^{[k]} = \boldsymbol{\Delta}^{[k]}\boldsymbol{A}_l^{[k]} + \boldsymbol{\Phi}^{[k]}\boldsymbol{B}_l^{[k]} \; . \end{aligned} \tag{6}$$

Since we already proved that (4) is a relaxation (3), it remains to show that (4) is equivalent to (6) (as defined in [19]). Note that this immediately follows from the one-to-one nature of the update in (5): indeed (5) should be seen as a one-to-one mapping $G$, from $\boldsymbol{U}_{l+1}^{[k]}$ to $\boldsymbol{A}_l^{[k]}, \boldsymbol{B}_l^{[k]}$ (for fixed $\boldsymbol{\Delta}^{[k]}$ and $\boldsymbol{\Phi}^{[k]}$), i.e., $G : \boldsymbol{U}_{l+1}^{[k]} \to G(\boldsymbol{A}_l^{[k]}, \boldsymbol{B}_l^{[k]})$.

Summarizing thus far, we relaxed the distributed IA problem in (3), but made the process of solving it more complex. In view of simplifying the solution process, we imposed a structure on the variables of the problem (the update rule in (5)): generally, this has the effect of constraining the variables to have a particular structure, i.e., adding an additional constraint set $\mathcal{S}$ to the problem. Thus $\mathcal{S}$ needs to be as "wide" as possible, such that it does not alter the feasible region. This is the reason for choosing a generic update rule (that results in $\mathcal{S}$ encompassing a "wide" range of matrices, e.g., unitary).

Although the relaxation argument implies that such a scheme will yield "better" solutions than its distributed IA counterpart, two comments on the latter statement are in order. Firstly, the obvious fact that the solution of the relaxed problem, (6), will be lower than that of the original problem, (3), is contingent upon both schemes being able to find the global solutions to their respective problems. Furthermore, since both problems have to be solved at every iteration, it is rather hard to show that at any given iteration, the leakage value for one of the schemes will be better or worse than the other one (since the sequence $\{\boldsymbol{Q}_l^{[k]}\}_l$ is different for each of the schemes). As a result, although the relaxation argument cannot lead to a rigorous proof of the superiority of any of the schemes, it does provide a well-founded heuristic for adopting such an update rule.

### B. Problem Formulation

Now that we showed that (6) is a relaxation of (3), we proceed to rewrite (6) into a simpler equivalent problem, making use of the following result.

**Proposition 1.** *Let $\boldsymbol{U} \in \mathbb{C}^{n\times p}, p < n$, be a given full rank matrix, and $\boldsymbol{Q} \in \mathcal{U}(n,p)$ a unitary matrix. Then there exists $\boldsymbol{A} \in \mathbb{C}^{p\times p}$ and $\boldsymbol{B} \in \mathbb{C}^{(n-p)\times p}$ such that $\boldsymbol{U} = \boldsymbol{Q}\boldsymbol{A} + \boldsymbol{Q}^{\perp}\boldsymbol{B}$, where $\boldsymbol{Q}^{\perp} \in \mathcal{U}(n,n-p)$. Furthermore, $\boldsymbol{A} = \boldsymbol{Q}^{\dagger}\boldsymbol{U}$ and $\boldsymbol{B} = \boldsymbol{Q}^{\perp\dagger}\boldsymbol{U}$.*

*Proof:* Refer to Appendix A ∎

As a result, Proposition 1 implies any $\boldsymbol{U}_{l+1}^{[k]} \in \mathbb{C}^{N\times d}$, can be written as $\boldsymbol{U}_{l+1}^{[k]} = \boldsymbol{\Delta}^{[k]}\boldsymbol{A}_l^{[k]} + \boldsymbol{\Phi}^{[k]}\boldsymbol{B}_l^{[k]}$, and consequently, the second constraint in (6) can be removed without changing the domain of the optimization problem. Then, by applying the one-to-one mapping $G : \boldsymbol{U}_{l+1}^{[k]} \to \boldsymbol{\Delta}^{[k]}\boldsymbol{A}_l^{[k]} + \boldsymbol{\Phi}^{[k]}\boldsymbol{B}_l^{[k]}$, we rewrite (6) as,

$$\begin{aligned} \min_{\boldsymbol{A}_l^{[k]}, \boldsymbol{B}_l^{[k]}} \; & f^{[k]}(\boldsymbol{A}_l^{[k]}, \boldsymbol{B}_l^{[k]}) = \mathrm{tr}[(\boldsymbol{\Delta}^{[k]}\boldsymbol{A}_l^{[k]} + \boldsymbol{\Phi}^{[k]}\boldsymbol{B}_l^{[k]})^{\dagger}\boldsymbol{Q}_{l+1}^{[k]} \\ & (\boldsymbol{\Delta}^{[k]}\boldsymbol{A}_l^{[k]} + \boldsymbol{\Phi}^{[k]}\boldsymbol{B}_l^{[k]})] \\ \text{s.t.} \; & \|\boldsymbol{A}_l^{[k]}\|_F^2 + \|\boldsymbol{B}_l^{[k]}\|_F^2 = P_r \; . \end{aligned} \tag{7}$$

[4] Generally speaking, there are other ways to "partition" the $N$-dimensional space in question, i.e., $\boldsymbol{\Delta}^{[k]} \in \mathcal{U}(N,r)$, $\boldsymbol{A}_l^{[k]} \in \mathbb{C}^{r\times d}$ and $\boldsymbol{\Phi}^{[k]} \in \mathcal{U}(N,N-r)$, $\boldsymbol{B}_l^{[k]} \in \mathbb{C}^{(N-r)\times d}$, where $1 \le r \le N-1$. However, in that case, selecting the best value of $r$ will likely depend on the particular problem instance, and thus will have to be selected based on empirical evidence. Consequently, we set $r = d$ for the sake of simplicity

### C. Turbo Optimization

Due to the fact that $f^{[k]}$ is not jointly convex in $\boldsymbol{A}_l^{[k]}$ and $\boldsymbol{B}_l^{[k]}$, alternately optimizing each of the variables stands out as a possible solution. Furthermore, even when one of the variables is fixed, the resulting optimization problem is still a non-convex one, due to the non-affine equality constraint. Still, it is possible to find the globally optimum solution for each of the variables, as shown in Lemma 1. By repeating this process multiple times, we wish to produce a *non-increasing sequence* $\{f^{[k]}(\boldsymbol{A}_{l,m}^{[k]}, \boldsymbol{B}_{l,m}^{[k]})\}_m$ ($m$ being the turbo iteration index) that converges to a non-negative limit. Thus, in addition to the main outer F-B iteration, $l$, we now have an inner loop (or turbo iteration), where $\boldsymbol{A}_{l,m}^{[k]}$ and $\boldsymbol{B}_{l,m}^{[k]}$ are sequentially optimized. With this in mind, for a given $\boldsymbol{B}_{l,m}^{[k]}$, the *sequential updates* $\boldsymbol{A}_{l,m+1}^{[k]}, \boldsymbol{B}_{l,m+1}^{[k]}$ are defined as follows,

$$\underbrace{\boldsymbol{B}_{l,m+1}^{[k]} \triangleq \underset{\boldsymbol{B}}{\text{argmin}}\ f^{[k]}\left(\underbrace{\boldsymbol{A}_{l,m+1}^{[k]} \triangleq \underset{\boldsymbol{A}}{\text{argmin}}\ f^{[k]}(\boldsymbol{A}, \boldsymbol{B}_{l,m}^{[k]})}_{J1}\ ,\ \boldsymbol{B}\right)}_{J2}\ ,$$

where the inner optimization problems are elaborated below,

$$(J1):\ \boldsymbol{A}_{l,m+1}^{[k]} = \underset{\boldsymbol{A}}{\text{argmin}}\ f^{[k]}(\boldsymbol{A}, \boldsymbol{B}_{l,m}^{[k]})$$
$$\text{s. t. } h_1(\boldsymbol{A}) = \ \|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}_{l,m}^{[k]}\|_F^2 - P_r = 0\ ,$$
$$(J2):\ \boldsymbol{B}_{l,m+1}^{[k]} = \underset{\boldsymbol{B}}{\text{argmin}}\ f^{[k]}(\boldsymbol{A}_{l,m+1}^{[k]}, \boldsymbol{B})$$
$$\text{s. t. } h_2(\boldsymbol{B}) = \|\boldsymbol{B}\|_F^2 + \|\boldsymbol{A}_{l,m+1}^{[k]}\|_F^2 - P_r = 0\ .$$

**Remark 1.** Both $(J1)$ and $(J2)$ are non-convex due to the quadratic equality constraint. Note that applying convex relaxation by replacing the equality by an inequality (thus forming a convex superset) will not help: indeed one can show in that that the sequences of optimal updates within the turbo iteration, are such that $\{\boldsymbol{A}_{l,m}^{[k]}\}_m \rightarrow \boldsymbol{0}$ and $\{\boldsymbol{B}_{l,m}^{[k]}\}_m \rightarrow \boldsymbol{0}$ (consequently, $\boldsymbol{U}_{l+1}^{[k]} = \boldsymbol{0}$ , implying that the algorithm converges to a point that does not necessarily correspond to stationary points of the leakage function).

The following lemma provides the solution to the different subproblems of our proposed algorithms.

**Lemma 1.** *Consider the following non-convex quadratic program,*

$$\min_{\boldsymbol{X}} f(\boldsymbol{X}) = \text{tr}[(\gamma_1\boldsymbol{\Theta} + \gamma_2\boldsymbol{T}\boldsymbol{X})^\dagger \boldsymbol{Q}(\gamma_1\boldsymbol{\Theta} + \gamma_2\boldsymbol{T}\boldsymbol{X})]$$
$$\text{s. t. } h(\boldsymbol{X}) = \|\boldsymbol{X}\|_F^2 - \zeta = 0\ ,\ \zeta > 0\ , \tag{8}$$

*where* $\boldsymbol{Q} \succeq \boldsymbol{0}$ , $\boldsymbol{\Theta} \neq \boldsymbol{0}$ , $0 \leq \gamma_1$ , $\gamma_2 \leq 1$. *Then, the (globally optimum) solution* $\boldsymbol{X}^\star$ *is given by*

$$\boldsymbol{X}^\star(\mu^\star) = -\gamma_1\gamma_2\left(\gamma_2^2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T} + \mu^\star\boldsymbol{I}\right)^{-1}\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{\Theta}\ , \tag{9}$$

*where* $\mu^\star$ *is the unique solution to*

$$\|\boldsymbol{X}^\star(\mu)\|_F^2 = \zeta$$

*in the interval* $-\gamma_2^2\ \lambda_1[\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T}] < \mu < \gamma_1\gamma_2\|\boldsymbol{\Theta}^\dagger\boldsymbol{Q}\boldsymbol{T}\|_F/\sqrt{\zeta}$. *Moreover,* $\|\boldsymbol{X}^\star(\mu)\|_F^2$ *is monotonically decreasing in* $\mu$, *for* $\mu > -\gamma_2^2\ \lambda_1[\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T}]$.

*Proof:* Refer to Appendix B. ■

Though it might seem that (4) can be solved using Lemma 1, i.e., by setting $\boldsymbol{\Theta} = \boldsymbol{0}$, this does make the necessary and sufficient conditions inconsistent (refer to Appendix B). On the other hand, it becomes clear at this point that $(J1)$ is a special case of (8), by letting $\boldsymbol{X} = \boldsymbol{A}$, $\boldsymbol{\Theta} = \boldsymbol{\Phi}^{[k]}\boldsymbol{B}_{l,m}^{[k]}$, $\boldsymbol{T} = \boldsymbol{\Delta}^{[k]}$, $\gamma_1 = \gamma_2 = 1$, $\zeta = P_r - \|\boldsymbol{B}_{l,m}^{[k]}\|_F^2$ (keeping in mind that $\|\boldsymbol{A}_{l,m}^{[k]}\|_F^2 + \|\boldsymbol{B}_{l,m}^{[k]}\|_F^2 = P_r$ , $\forall m$, it is evident that $\zeta > 0$). Applying the result of Lemma 1, we now write the solution to $(J1)$ as,

$$\boldsymbol{A}_{l,m+1}^{[k]}(\mu) = -(\boldsymbol{\Delta}^{[k]^\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}^{[k]} + \mu\boldsymbol{I})^{-1}\boldsymbol{\Delta}^{[k]^\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Phi}^{[k]}\boldsymbol{B}_{l,m}^{[k]},$$
$$\mu \in \{\ \mu \mid g(\mu) = \|\boldsymbol{A}_{l,m+1}^{[k]}(\mu)\|_F^2 + \|\boldsymbol{B}_{l,m}^{[k]}\|_F^2 - P_r = 0\ ,$$
$$\mu > -\lambda_1[\boldsymbol{\Delta}^{[k]^\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}^{[k]}]\}. \tag{10}$$

Since the function $g(\mu)$ is monotonically decreasing, the solution can be efficiently found using bisection.

The process of solving $(J2)$ follows exactly the same reasoning as above. By letting $\boldsymbol{X} = \boldsymbol{B}$, $\boldsymbol{\Theta} = \boldsymbol{\Delta}^{[k]}\boldsymbol{A}_{l,m+1}^{[k]}$, $\boldsymbol{T} = \boldsymbol{\Phi}^{[k]}$, $\gamma_1 = \gamma_2 = 1$, $\zeta = P_r - \|\boldsymbol{A}_{l,m+1}^{[k]}\|_F^2$ , $\zeta > 0$. Then, the application of of Lemma 1 immediately yields the solution to $(J2)$,

$$\boldsymbol{B}_{l,m+1}^{[k]}(\mu) = -(\boldsymbol{\Phi}^{[k]^\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Phi}^{[k]} + \mu\boldsymbol{I})^{-1}\boldsymbol{\Phi}^{[k]^\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}^{[k]}\boldsymbol{A}_{l,m+1}^{[k]},$$
$$\mu \in \{\ \mu \mid g(\mu) = \|\boldsymbol{B}_{l,m+1}^{[k]}(\mu)\|_F^2 + \|\boldsymbol{A}_{l,m+1}^{[k]}\|_F^2 - P_r = 0\ ,$$
$$\mu > -\lambda_1[\boldsymbol{\Phi}^{[k]^\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Phi}^{[k]}]\}. \tag{11}$$

### D. Reverse network optimization

Due to the inherent nature of the leakage function, the reverse network optimization follows the same reasoning as the one presented above. Thus, to avoid unnecessary repetition, we just limit ourselves to stating the results, skipping all the derivations. The update rule for the transmit filter as is set as follows (similarly to (5)),

$$\boldsymbol{V}_{l+1}^{[k]} = \boldsymbol{\Lambda}^{[k]}\boldsymbol{C}_l^{[k]} + \boldsymbol{\Gamma}^{[k]}\boldsymbol{D}_l^{[k]}, \tag{12}$$

where $\boldsymbol{\Lambda}^{[k]} \in \mathcal{U}(M, d)$ and $\boldsymbol{\Gamma}^{[k]} \in \mathcal{U}(M, M-d)$ are such that $\boldsymbol{\Lambda}^{[k]^\dagger}\boldsymbol{\Gamma}^{[k]} = \boldsymbol{0}$. Furthermore, $\boldsymbol{C}_l^{[k]} \in \mathbb{C}^{d\times d}$ and $\boldsymbol{D}_l^{[k]} \in \mathbb{C}^{(M-d)\times d}$ are the combining weights of $\boldsymbol{\Lambda}^{[k]}$ and $\boldsymbol{\Gamma}^{[k]}$, respectively. Then, the resulting sequential optimization problems are given as follows,

$$(J3):\ \boldsymbol{C}_{l,m+1}^{[k]} = \underset{\boldsymbol{C}}{\text{argmin}}\ \bar{f}^{[k]}(\boldsymbol{C}, \boldsymbol{D}_{l,m}^{[k]})$$
$$\text{s. t. } h_3(\boldsymbol{C}) = \ \|\boldsymbol{C}\|_F^2 + \|\boldsymbol{D}_{l,m}^{[k]}\|_F^2 - P_t = 0,$$
$$(J4):\ \boldsymbol{D}_{l,m+1}^{[k]} = \underset{\boldsymbol{D}}{\text{argmin}}\ \bar{f}^{[k]}(\boldsymbol{C}_{l,m+1}^{[k]}, \boldsymbol{D})$$
$$\text{s. t. } h_4(\boldsymbol{D}) = \|\boldsymbol{D}\|_F^2 + \|\boldsymbol{C}_{l,m+1}^{[k]}\|_F^2 - P_t = 0,$$

where $P_t$ is the transmit filter power constraint, and $\bar{f}^{[k]}$, the leakage at transmitter $k$, is given by

$$\bar{f}^{[k]}(\boldsymbol{C}_l^{[k]}, \boldsymbol{D}_l^{[k]}) = \text{tr}(\boldsymbol{V}_{l+1}^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{V}_{l+1}^{[k]})$$
$$= \text{tr}[(\boldsymbol{\Lambda}^{[k]}\boldsymbol{C}_l^{[k]} + \boldsymbol{\Gamma}^{[k]}\boldsymbol{D}_l^{[k]})^\dagger\bar{\boldsymbol{Q}}_{l+1}^{[k]}(\boldsymbol{\Lambda}^{[k]}\boldsymbol{C}_l^{[k]} + \boldsymbol{\Gamma}^{[k]}\boldsymbol{D}_l^{[k]})].$$

Again, the same block coordinate descent structure can be employed to optimize the weight matrices $\boldsymbol{C}_l^{[k]}$ and $\boldsymbol{D}_l^{[k]}$. Using the same reasoning as earlier, one can obtain the optimal updates, using the result of Lemma 1, to yield,

$$\begin{aligned} &\boldsymbol{C}_{l,m+1}^{[k]}(\mu) = -(\boldsymbol{\Lambda}^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Lambda}^{[k]} + \mu\boldsymbol{I})^{-1}\boldsymbol{\Lambda}^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Gamma}^{[k]}\boldsymbol{D}_{l,m}^{[k]}, \\ &\mu \in \{\ \mu \mid g(\mu) = \|\boldsymbol{C}_{l,m+1}^{[k]}(\mu)\|_F^2 + \|\boldsymbol{D}_{l,m}^{[k]}\|_F^2 - P_t = 0\ , \\ &\qquad \mu > -\lambda_1[\boldsymbol{\Lambda}^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Lambda}^{[k]}]\}, \end{aligned} \tag{13}$$

$$\begin{aligned} &\boldsymbol{D}_{l,m+1}^{[k]}(\mu) = -(\boldsymbol{\Gamma}^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Gamma}^{[k]} + \mu\boldsymbol{I})^{-1}\boldsymbol{\Gamma}^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Lambda}^{[k]}\boldsymbol{C}_{l,m+1}^{[k]}, \\ &\mu \in \{\ \mu \mid g(\mu) = \|\boldsymbol{D}_{l,m+1}^{[k]}(\mu)\|_F^2 + \|\boldsymbol{C}_{l,m+1}^{[k]}\|_F^2 - P_t = 0\ , \\ &\qquad \mu > -\lambda_1[\boldsymbol{\Gamma}^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Gamma}^{[k]}]\}. \end{aligned} \tag{14}$$

### E. Convergence Analysis

As shown earlier, although the problem solved within the turbo iteration is a non-convex one (7), we can still show that the application of the updates for $\boldsymbol{A}_{l,m+1}^{[k]}$ and $\boldsymbol{B}_{l,m+1}^{[k]}$ (given in (10) and (11), respectively), cannot increase the leakage at each receiver (7).

**Theorem 1.** *For fixed $\boldsymbol{V}_{l+1}^{[1]}, ..., \boldsymbol{V}_{l+1}^{[K]}$, the leakage within the receiver turbo iteration is non-increasing, i.e. the sequence $\{f^{[k]}(\boldsymbol{A}_{l,m}^{[k]}, \boldsymbol{B}_{l,m}^{[k]})\}_m$ is non-increasing, and converges to a non-negative limit $f_{l,st}^{[k]} \geq 0$, where $\boldsymbol{A}_{l,m+1}^{[k]}$ and $\boldsymbol{B}_{l,m+1}^{[k]}$ are given in (10) and (11).*

*Proof:* The proof immediately follows from showing that for a fixed F-B iteration number $l$, the following holds,

$$f^{[k]}(\boldsymbol{A}_{l,m+1}^{[k]}, \boldsymbol{B}_{l,m+1}^{[k]}) \overset{(b)}{\leq} f^{[k]}(\boldsymbol{A}_{l,m+1}^{[k]}, \boldsymbol{B}_{l,m}^{[k]}) \overset{(a)}{\leq} f^{[k]}(\boldsymbol{A}_{l,m}^{[k]}, \boldsymbol{B}_{l,m}^{[k]}), \forall m. \tag{15}$$

Note that $(a)$ follows immediately from the definition and solution of $(J1)$. Consequently, the application of the update $\boldsymbol{A}_{l,m}^{[k]} \leftarrow \boldsymbol{A}_{l,m+1}^{[k]}$, given by (10), cannot increase the cost function. Similarly, points that satisfy (11) mimimize $(J2)$ (as shown by Lemma 1). Thus, the update $\boldsymbol{B}_{l,m}^{[k]} \leftarrow \boldsymbol{B}_{l,m+1}^{[k]}$ given in (11) cannot increase the cost function, and $(b)$ follows. Therefore, the sequence $\{f^{[k]}(\boldsymbol{A}_{l,m}^{[k]}, \boldsymbol{B}_{l,m}^{[k]})\}_m$ is non-increasing, and since the leakage function is non-negative, we conclude that $\{f^{[k]}(\boldsymbol{A}_{l,m}^{[k]}, \boldsymbol{B}_{l,m}^{[k]})\}_m$ converges to some non-negative limit $f_{l,st}^{[k]}$. ■

With this in mind, not only does Theorem 1 establish the convergence of the turbo iteration to some limit, but also that the leakage is non-increasing with each of the updates (as immediately seen from (15)). Although Theorem 1 shows the convergence of the turbo iteration, to some limit, one cannot claim that this limit corresponds to a stationary point of the function, because the variables in (7) are coupled [20]. Moreover, recall that we do not wish our algorithm to converge to stationary points of the leakage function since the latter correspond to rank-one solutions (following the discussion in Sect. II-B). Consequently, showing the convergence of the block coordinate descent method to stationary points becomes much less critical in our case, as long as we can establish the non-increasing nature of the leakage. In addition, it is not hard to see that exactly the same reasoning can be used to extrapolate the result of Theorem 1 to show that the updates for the transmit filter weights (given in (13) and (14)), can only decrease the leakage at the given transmitter, and thus establishing the convergence of the turbo iteration for the transmit filter weights.

### F. Convergence to lower-rank solutions

For convenience, we define $L$ as the maximum number of F-B iterations, and $T$ as the maximum number of turbo iterations, for our algorithm. Strong (empirical) evidence suggests that *the proposed algorithm will gradually reduce the transmit / receive filter rank, and converge to rank-one solutions, as $L, T \to \infty$*. As a result, operating the algorithm with large values of $L, T$ will result in a multiplexing gain of 1 degree-of-freedom per user (highly suboptimal especially if multistream transmission is desired). Conversely, by allowing the algorithm to gradually reduce the rank of a given transmit / receive filter[5], we exploit the "transient phase" of this algorithm stopping before convergence to rank-one solutions (i.e. for small values of $L, T$). In addition, recall that reducing the transmit / receive filter rank also reduces the dimension of the interference that is caused to other receivers (this is beneficial in the interference-limited regime): this makes the alignment of interference "easier" and greatly speeds up the convergence. Note as well that although having small values of $L, T$ is extremely desirable (the associated communication and computational overhead will be relatively low), having them too small will evidently result in poor performance, e.g., $L = 0, T = 0$. This does suggest the existence of a trade-off on $L$ and $T$, between the performance and overhead. Unfortunately, a mathematical characterization of the latter reveals to be impossible, and we will rely on empirical evidence to select them.

---

**Algorithm 1** Iterative Weight Update with Rank-Reduction (IWU-RR)

---

**for** $l = 0, 1, ..., L-1$ **do**
  *// forward network optimization*
  Update receiver interference covariance matrix
  **for** $m = 0, 1, ..., T-1$ **do**
    Compute $\{\boldsymbol{A}_{l,m+1}^{[k]}\}_k$ in (10), $\{\boldsymbol{B}_{l,m+1}^{[k]}\}_k$ in (11)
  **end for**
  Check rank and perform rank-reduction [5]
  Update receive filter in (5)
  *// reverse network optimization*
  Update transmitter interference covariance matrix
  **for** $m = 0, 1, ..., T-1$ **do**
    Compute $\{\boldsymbol{C}_{l,m+1}^{[k]}\}_k$ in (13), $\{\boldsymbol{D}_{l,m+1}^{[k]}\}_k$ in (14)
  **end for**
  Check rank and perform rank-reduction [5]
  Update transmit precoder in (12)
**end for**

---

[5] If the weight combining matrices at the output of the turbo iteration (for, say, the receive filter update) are rank deficient, then resulting receive filter is rank deficient as well. The rank-reduction process is done by eliminating linearly dependent columns of $\boldsymbol{A}_{l,T}^{[k]}$ and $\boldsymbol{B}_{l,T}^{[k]}$, and appropriately scaling each of them, to fulfil the power constraint.

## IV. Rank-Preserving Updates

### A. Proposed Update Rule and Problem Formulation

An inherent consequence of the coupled nature of the weight updates for $\boldsymbol{A}_l^{[k]}$ and $\boldsymbol{B}_l^{[k]}$, i.e., (10) and (11) (as well as the turbo-like structure of the algorithm), is the fact that if any of the latter are rank-deficient, then the other one will be rank-deficient as well. Moreover, imposing an explicit rank constraint would make the problem extremely hard to solve (since most rank-constrained problems are NP-hard). Alternately, one way to have the algorithm yield full-rank solutions, is to use another update rule (shown below) where this effect is absent, i.e.,

$$\boldsymbol{U}_{l+1}^{[k]} = \sqrt{1-\beta_l^{[k]2}}\boldsymbol{U}_l^{[k]} + \beta_l^{[k]}\boldsymbol{\Delta}_l^{[k]}\boldsymbol{Z}_l^{[k]},\ 0 \le \beta_l^{[k]} \le 1, \quad (16)$$

where $\boldsymbol{\Delta}_l^{[k]} \in \mathcal{U}(N, N-d)$ is such that $\boldsymbol{\Delta}_l^{[k]} \subseteq (\boldsymbol{U}_l^{[k]})^{\perp}$, $\boldsymbol{Z}_l^{[k]} \in \mathbb{C}^{(N-d)\times d}$ is the combining weight matrix for the receiver update, and $\beta_l^{[k]}$ is the step size for the receive filter update. Note that due to the dependence of the update on the current receive filter, $\boldsymbol{U}_l^{[k]}$, it is easy to verify that $\boldsymbol{U}_{l+1}^{[k]}$ is full rank, if $\boldsymbol{U}_l^{[k]}$ is. In addition, if both $\boldsymbol{U}_l^{[k]}$ and $\boldsymbol{Z}_l^{[k]}$ satisfy the power constraint, i.e., $\|\boldsymbol{U}_l^{[k]}\|_F^2 = P_r$ and $\|\boldsymbol{Z}_l^{[k]}\|_F^2 = P_r$, then $\|\boldsymbol{U}_{l+1}^{[k]}\|_F^2 = P_r$.

Similarly to (6), by incorporating the above update structure, the resulting optimization problem at each receiver is stated as follows,

$$\begin{aligned}\min_{\boldsymbol{U}_{l+1}^{[k]}}\ & f^{[k]}(\boldsymbol{U}_{l+1}^{[k]}) = \text{tr}(\boldsymbol{U}_{l+1}^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{U}_{l+1}^{[k]})\\ \text{s.t. }& \|\boldsymbol{U}_{l+1}^{[k]}\|_F^2 = P_r\\ & \boldsymbol{U}_{l+1}^{[k]} = \sqrt{1-\beta_l^{[k]2}}\boldsymbol{U}_l^{[k]} + \beta_l^{[k]}\boldsymbol{\Delta}_l^{[k]}\boldsymbol{Z}_l^{[k]}. \quad (17)\end{aligned}$$

A few comments are in order at this point regarding the similarities and fundamental differences between the rank-reducing update proposed earlier, and the rank-preserving update above. Given that both result in non-convex optimization problems, they both rely on a coordinate descent approach to optimize each of their respective variables. In addition, it is clear that the rank-reducing update in (5) is more generic than the rank-preserving update in (16). As a result, the relaxation argument that was put forth to motivate the use of the update in (5) (Sect. III-A), no longer holds here. Furthermore, both algorithms have exactly the same structure: in that sense, after updating its interference covariance matrix, receiver $k$ wishes to optimize both its combining weight and step-size, i.e. $\beta_l^{[k]}$ and $\boldsymbol{Z}_l^{[k]}$, such as to minimize the resulting interference leakage at the next iteration. Plugging (16) into (17) yields the cost function at receiver $k$,

$$\begin{aligned}f^{[k]}(\beta_l^{[k]},\boldsymbol{Z}_l^{[k]}) &= (1-\beta_l^{[k]^2})\text{tr}(\boldsymbol{U}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{U}_l^{[k]})\\ &+ \beta_l^{[k]^2}\text{tr}(\boldsymbol{Z}_l^{[k]\dagger}\boldsymbol{\Delta}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}_l^{[k]}\boldsymbol{Z}_l^{[k]})\\ &+ 2\beta_l^{[k]}\sqrt{1-\beta_l^{[k]2}}\,Re[\text{tr}(\boldsymbol{U}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}_l^{[k]}\boldsymbol{Z}_l^{[k]})]. \quad (18)\end{aligned}$$

### B. Inner Optimization

Again, we will use block coordinate descent to mitigate the non-convexity of (18), implying that receiver $k$ optimizes both its weight combining matrix and step size ($\boldsymbol{Z}_l^{[k]}$ and $\beta_l^{[k]}$), alternately and sequentially, within the turbo iteration, to produce a non-increasing sequence $\{f^{[k]}(\beta_{l,m}^{[k]},\boldsymbol{Z}_{l,m}^{[k]})\}_m$ that will converge to some non-negative limit. Thus, given $\beta_{l,m}^{[k]}$ at the $m^{th}$ turbo iteration, the *sequential* updates $\boldsymbol{Z}_{l,m+1}^{[k]}$ and $\beta_{l,m+1}^{[k]}$ are chosen, as follows,

$$\underbrace{\beta_{l,m+1}^{k} \triangleq \underset{\beta}{\text{argmin}}\ f^{[k]}\left(\beta,\ \underbrace{\boldsymbol{Z}_{l,m+1}^{[k]} \triangleq \underset{\boldsymbol{Z}}{\text{argmin}}\ f^{[k]}(\beta_{l,m}^{k},\boldsymbol{Z})}_{K1}\right)}_{K2},$$

$$\text{where } (K1):\ \boldsymbol{Z}_{l,m+1}^{[k]} = \underset{\boldsymbol{Z}}{\text{argmin}}\ f^{[k]}(\beta_{l,m}^{[k]},\boldsymbol{Z}) \quad \text{s.t. } h_1(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_F^2 = P_r.$$

Note that $(K1)$ in non-convex due the quadratic equality constraint, but can be solved using Lemma 1 by letting $\boldsymbol{X} = \boldsymbol{Z}$, $\boldsymbol{\Theta} = \boldsymbol{U}_l^{[k]}$, $\boldsymbol{T} = \boldsymbol{\Delta}_l^{[k]}$, $\gamma_1 = \sqrt{1-\beta_{l,m}^{[k]2}}$ , $\gamma_2 = \beta_{l,m}^{[k]}$, $\zeta = P_r$. Applying the result of Lemma 1 the optimal update is given by,

$$\begin{aligned}&\boldsymbol{Z}_{l,m+1}^{[k]}(\mu) = -\beta_{l,m}^{[k]}\sqrt{1-\beta_{l,m}^{[k]2}}\\ &\qquad\left(\beta_{l,m}^{[k]^2}\boldsymbol{\Delta}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}_l^{[k]} + \mu I\right)^{-1}\boldsymbol{\Delta}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{U}_l^{[k]},\\ &\mu \in \{\ \mu \mid g(\mu) = \|\boldsymbol{Z}_{l,m+1}^{[k]}(\mu)\|_F^2 - P_r = 0,\\ &\qquad \mu > -\beta_{l,m}^{[k]^2}\lambda_1[\boldsymbol{\Delta}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}_l^{[k]}]\}. \quad (19)\end{aligned}$$

Given $\boldsymbol{Z}_{l,m+1}^{[k]}$, the optimization for $\beta_{l,m}^{[k]}$ is formulated as follows,

$$\begin{aligned}(K2):\beta_{l,m+1}^{[k]} = \ &\underset{\beta}{\text{argmin}}\ f^{[k]}(\beta,\boldsymbol{Z}_{l,m+1}^{[k]}) = (1-\beta^2)e_1\\ &+ \beta\sqrt{1-\beta^2}e_2 + \beta^2 e_3\\ &\text{s.t. } \ 0 \le \beta \le 1, \quad (20)\end{aligned}$$

where we let $e_1 = \text{tr}(\boldsymbol{U}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{U}_l^{[k]})$, $e_2 = 2Re[\text{tr}(\boldsymbol{U}_l^{[k]\dagger}\boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}_l^{[k]}\boldsymbol{Z}_{l,m+1}^{[k]})]$, $e_3 = \text{tr}(\boldsymbol{Z}_{l,m+1}^{[k]\dagger}\boldsymbol{\Delta}_l^{[k]\dagger}\ \boldsymbol{Q}_{l+1}^{[k]}\boldsymbol{\Delta}_l^{[k]}\boldsymbol{Z}_{l,m+1}^{[k]})$, for notational simplicity.

The main issue that one has to carefully consider while optimizing $\beta_{l,m}^{[k]}$ is that the sign and magnitude of $e_2$ in (20) may vary depending on the particular instance and channel realization. Furthermore, we also need to rule out the fact that $f^{[k]}$ might in fact be concave in $\beta_{l,m}^{[k]}$ (since by finding the stationary points, we would be maximizing our cost function), or having many extrema. The result of Lemma 2 addresses all those issues (whose proof is given in Appendix C).

**Lemma 2.** *The function* $p(x) = (1-x^2)e_1 + x\sqrt{1-x^2}e_2 + x^2e_3$ *is convex on the interval* $[0,\ 1]$*, and thus has a single unique global minimum given by* $x^* = \left(\frac{1}{2} + \frac{e_1-e_3}{2\sqrt{(e_1-e_3)^2+e_2^2}}\right)^{1/2}$.

*Proof:* Refer to Appendix C. ■

Lemma 2 establishes the uniqueness of the solution to $(K2)$, by showing that $f^{[k]}(\beta_{l,m}^{[k]},\boldsymbol{Z}_{l,m+1}^{[k]})$ is indeed convex in $\beta_{l,m}^{[k]}$.

Thus, the update for $\beta_{l,m}^{[k]}$ can be simply expressed as,

$$\beta_{l,m+1}^{[k]} = x^* = \left(\frac{1}{2} + \frac{e_1 - e_3}{2\sqrt{(e_1-e_3)^2 + e_2^2}}\right)^{1/2}. \quad (21)$$

### C. Reverse Network Optimization

We again exploit the duality that is inherent to the structure of the leakage function, to apply the same reasoning to obtain the optimal updates for the reverse network optimization phase. Thus, skipping all the details, we limit ourselves to just presenting the results. Similarly to the receiver update, each transmitter updates its filter according to the following rule,

$$\boldsymbol{V}_{l+1}^{[k]} = \sqrt{1-\alpha_l^{[k]^2}}\boldsymbol{V}_l^{[k]} + \alpha_l^{[k]}\boldsymbol{\Phi}_l^{[k]}\boldsymbol{W}_l^{[k]},\ 0 \le \alpha_l^{[k]} \le 1, \quad (22)$$

where $\boldsymbol{\Phi}_l^{[k]} \in \mathcal{U}(M, M-d)$ is such that $\boldsymbol{\Phi}_l^{[k]} \in (\boldsymbol{V}_l^{[k]})^{\perp}$, and $\boldsymbol{W}_l^{[k]} \in \mathbb{C}^{(M-d)\times d}$ is the matrix of combining weights. Thus, the resulting optimization problems solved within the turbo iteration are as follows,

$$(K3): \boldsymbol{W}_{l,m+1}^{[k]} = \underset{\boldsymbol{W}}{\text{argmin}}\ \bar{f}^{[k]}(\alpha_{l,m}^{[k]}, \boldsymbol{W}) \quad \text{s.t.}\ \ h_2(\boldsymbol{W}) = \|\boldsymbol{W}\|_F^2 = P_t,$$

$$(K4): \alpha_{l,m+1}^{[k]} = \underset{\alpha}{\text{argmin}}\ \bar{f}^{[k]}(\alpha, \boldsymbol{W}_{l,m+1}^{[k]}) \quad \text{s.t.}\ \ 0 \le \alpha \le 1$$

where the interference leakage at transmitter $k$ is given by,

$$\begin{aligned}\bar{f}^{[k]}(\alpha_{l,m}^{[k]}, \boldsymbol{W}_{l,m}^{[k]}) &= (1-\alpha_{l,m}^{[k]^2})\text{tr}(\boldsymbol{V}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{V}_l^{[k]}) \\ &+ \alpha_{l,m}^{[k]^2}\text{tr}(\boldsymbol{W}_{l,m}^{[k]^\dagger}\boldsymbol{\Phi}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Phi}_l^{[k]}\boldsymbol{W}_{l,m}^{[k]}) \\ &+ 2\alpha_{l,m}^{[k]}\sqrt{1-\alpha_{l,m}^{[k]^2}}Re[\text{tr}(\boldsymbol{V}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Phi}_l^{[k]}\boldsymbol{W}_{l,m}^{[k]})]. \end{aligned} \quad (23)$$

Finally, the optimal updates within the turbo iteration are as follows,

$$\begin{aligned}\boldsymbol{W}_{l,m+1}^{[k]}(\mu) &= -\alpha_{l,m}^{[k]}\sqrt{1-\alpha_{l,m}^{[k]^2}} \\ &\left(\alpha_{l,m}^{[k]^2}\boldsymbol{\Phi}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Phi}_l^{[k]} + \mu I\right)^{-1}\boldsymbol{\Phi}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{V}_l^{[k]}, \\ \mu \in \{\ \mu\ |\ &\|\boldsymbol{W}_{l,m+1}^{[k]}(\mu)\|_F^2 - P_t = 0, \\ &\mu > -\alpha_{l,m}^{[k]^2}\lambda_1[\boldsymbol{\Phi}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Phi}_l^{[k]}]\}. \end{aligned} \quad (24)$$

Using Lemma 2, the optimal update for $\alpha_{l,m}^{[k]}$ is,

$$\alpha_{l,m}^{[k]} = \left(\frac{1}{2} + \frac{b_1 - b_3}{2\sqrt{(b_1-b_3)^2 + b_2^2}}\right)^{1/2}, \quad (25)$$

where $b_1 = \text{tr}(\boldsymbol{V}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{V}_l^{[k]})$, $b_2 = 2Re[\text{tr}(\boldsymbol{V}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Phi}_l^{[k]}\boldsymbol{W}_{l,m+1}^{[k]})]$, $b_3 = \text{tr}(\boldsymbol{W}_{l,m+1}^{[k]^\dagger}\boldsymbol{\Phi}_l^{[k]^\dagger}\bar{\boldsymbol{Q}}_{l+1}^{[k]}\boldsymbol{\Phi}_l^{[k]}\boldsymbol{W}_{l,m+1}^{[k]})$.

### D. Convergence of turbo iteration

The convergence of the turbo iteration (for both the receive and transmit filter updates) can be established using a similar reasoning as the one used in Sect III-E. In other words, we show that the application of each update cannot increase the cost function, i.e.,

$$f^{[k]}(\beta_{l,m+1}^{[k]}, \boldsymbol{Z}_{l,m+1}^{[k]}) \overset{(b)}{\le} f^{[k]}(\beta_{l,m}^{[k]}, \boldsymbol{Z}_{l,m+1}^{[k]}) \overset{(a)}{\le} f^{[k]}(\beta_{l,m}^{[k]}, \boldsymbol{Z}_{l,m}^{[k]}).$$

**Algorithm 2** Iterative Weight Update with Rank-Preservation (IWU-RP)

```
for l = 0, 1, ..., L − 1 do
    // forward network optimization
    Update receiver interference covariance matrix
    for m = 0, 1, ..., T − 1 do
        Compute {Z^[k]_{l,m+1}}_k in (19), {β^[k]_{l,m+1}}_k in (21)
    end for
    Update receive filter in (16)
    // reverse network optimization
    Update transmitter interference covariance matrix
    for m = 0, 1, ..., T − 1 do
        Compute {W^[k]_{l,m+1}}_k in (24), {α^[k]_{l,m+1}}_k in (25)
    end for
    Update transmit precoder in (12)
end for
```

The proof follows exactly the same argument in as the one in Theorem 1, i.e., by showing that the updates $\boldsymbol{Z}_{l,m}^{[k]} \leftarrow \boldsymbol{Z}_{l,m+1}^{[k]}$ in (19), and $\beta_{l,m}^{[k]} \leftarrow \beta_{l,m+1}^{[k]}$ in (21) cannot increase the cost function.

## V. Implementation Aspects and Complexity

The major drawback for previously proposed distributed schemes that rely on F-B iterations, is that they assume a large number of F-B iterations to deliver their intended performance, ranging from hundreds to thousands (as we shall see in the next section) - a prohibitively high cost since they correspond to actual channel uses between the transmitter and receiver. The chief advantage of the proposed approach is the fact that it greatly reduces the latter communication overhead to a few iterations, while still retaining a very high performance (as simulations will show).

We will use the flop count as a surrogate measure of complexity, although it is well known that the latter is a rather coarse one. Assume for simplicity that $d = N/2 = M/2 \triangleq n$ (this is consistent with the simulation parameters), and denote by $\mathcal{C}$ the complexity per F-B iteration. Note that the latter quantity will be largely dominated by the computationally demanding operations such as matrix product, matrix inversion, and eigenvalue decomposition (EVD). With this in mind, for $\boldsymbol{X} \in \mathbb{C}^{m\times p}, \boldsymbol{Y} \in \mathbb{C}^{p\times n}$, then $\boldsymbol{XY}$ needs $8mnp$ flops. Furthermore, inverting an $n\times n$ matrix requires $2n^3 - n$ flops, while computing the EVD of an $n \times n$ Hermitian matrix using the SVD requires $126n^3$ flops[6], resulting in each update in IWU-RR requiring $(67/4)n^3 - n$. Thus, keeping in mind that each iteration involves $2K$ such updates repeated $T$ times, and that EVD is applied to an $n/2 \times n/2$ matrix, the complexity of IWU-RR is

$$\mathcal{C}_{IWU-RR} = (2K)(T)(16n^3 + 67n^3 - 4n) = 2KT(83n^3 - 4n).$$

[6]Generally speaking, the complexity of operations such EVD or SVD, are data dependent: though it is well-known that they are $\mathcal{O}(n^3)$, the exact values depend on the matrix itself. For simplicity, we approximate the complexity of an $n \times n$ SVD as $126n^3$ [21].

The same logic applies in the case of IWU-RP, except that each update requires $(53/4n^3 - n)$, to yield

$$\mathcal{C}_{IWU-RP} = 2KT(95/2n^3 - n).$$

Given that the complexity of the bisection method is negligible in comparison with the above, and that the latter depends on the channel realization, and many of the problem parameters (making it extremely difficult to characterize), we have ignored the cost of the bisection method in both cases. Finally, for distributed IA, the cost is largely dominated by the EVD of an $n \times n$ matrix, to yield

$$\mathcal{C}_{DIA} = 2K(126n^3).$$

Since our schemes employ relatively small values of $T$, the complexity (per F-B iteration) is similar for all the above schemes (albeit slightly lower for distributed IA). However, our simulations generally indicated that our proposed algorithms require a much smaller number of F-B iterations to reach a predetermined tolerance value. Consequently, the overall complexity of our schemes will be much lower.

## VI. Simulation Results

As stated earlier, the performance of the proposed schemes is largely dependent on the number of F-B iterations $L$, as well as the number of turbo iterations $T$. Since any explicit optimization of the latter quantities is a rather tedious task - if not infeasible, we will rely on simulations to evaluate their effect, as well as both algorithms' performance. For that matter, we fix the maximum number of F-B iterations to a small value, e.g., $L = 2$ (since we wish to keep the communication overhead at a low level), and evaluate the algorithms' performance for several values of $T$. In addition, initializing distributed IA with random rank-$d$ unitary transmit filters, the stopping criterion in all subsequent simulations is a maximum number of F-B iterations $L$, thus keeping the overhead the same for all schemes. Although in this case, the proposed schemes will have higher computational overhead with respect to distributed IA, this will easily be offset by the gains in performance (as this section will clearly show).

We choose the matrices $\{\mathbf{\Delta}^{[k]}\}, \{\mathbf{\Phi}^{[k]}\}$ (for the receive filter optimization), and $\{\mathbf{\Lambda}^{[k]}\}, \{\mathbf{\Gamma}^{[k]}\}$ (for the transmit filter optimization) as random unitary matrices obtained by applying the QR decomposition to random matrices with Gaussian i.i.d entries. Because the latter matrices are fixed throughout the entire algorithm, we can see that their choice is irrelevant, firstly since it is not based on some a priori channel information (for instance, the performance will improve by choosing $\{\mathbf{\Delta}^{[k]}\}$ to span the range of $\boldsymbol{H}^{[kk]}$ ). Moreover, we generate the channel matrices as i.i.d. circular Gaussian random variables, which are stochastically invariant to unitary tranformations. All the sum-rate curves are averaged over 1000 channel realizations. We reiterate the important fact that our schemes only optimize the interference subspace, without any regard to the signal or noise. Thus, a comparison with schemes such as max-SINR [2] and (weighted) MMSE [5], [6] is somewhat not relevant for this work, since they also optimize the desired signal subspace.

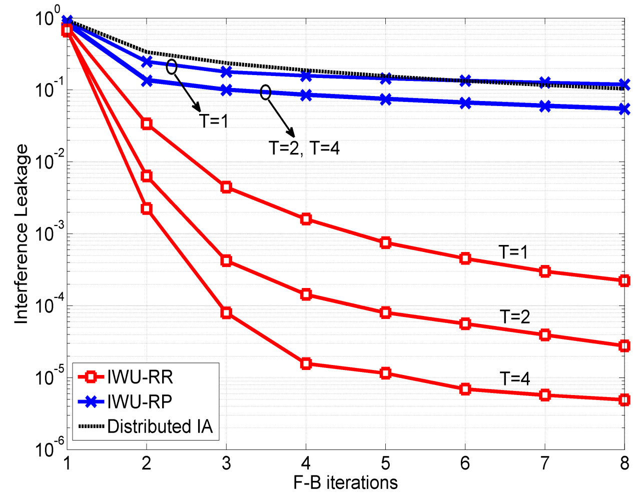


Fig. 2: Interference Leakage as a function of $L$, $T$ ($4 \times 4$ MIMO, 4 users, $d = 2$)

### A. Evolution of Interference Leakage versus L and T

Using insights from the feasibility of IA [10], [22] , we test the robustness of the proposed schemes against the following scenario, known a priori to be infeasible. Though this might seem to put distributed IA at a disadvantage (given that the latter is designed to handle feasible scenarios), scenarios that are known to be feasible are few, and might not always be of practical interest. Thus, robustness to infeasible IA configurations is desirable. Fig. 2 shows the (average) evolution of the leakage with the number of F-B iterations, for both our schemes (plotted for several values of $T$), and distributed IA. Although both schemes outperform distributed IA for any value of $T$, *the gap between IWU-RR and the benchmark is indeed impressive* ($\sim 3$ to 5 orders of magnitude, depending on the value of $T$). As expected, this gain stems from the ability of IWU-RR to perform rank-reduction, thereby decreasing the dimension of the interference at the corresponding receiver. In addition we observe that the gain from each additional turbo iteration is decreasing: this is clearly visible in the case of IWU-RP, where the curves corresponding to $T = 2$ and $T = 4$ are almost identical, implying that *only a few turbo iteration are needed to give the desired performance boost.*

### B. Sum-Rate Performance

Next, we simulate the ergodic sum-rate of both our schemes for a $10 \times 10$ MIMO, 4-user MIMO IC with $d = 4$, known to be proper [22], and fix the number of F-B iterations to 2, for all algorithms. We use the distributed IA algorithm in [2] as a benchmark, but most importantly, we also include the rank-one solution to (4), given by SDR. Fig. 3 reflects the effect of the turbo iteration on the sum-rate performance of both algorithms: *by running just a few turbo iterations, we see that both schemes significantly outperform distributed IA, especially in the high SNR region, when the gain becomes very large!* In addition we observe that indeed the rank-one solution of SDR offers extremely poor performance in terms of sum-rate (as discussed in Sect. II.B). Moreover, we

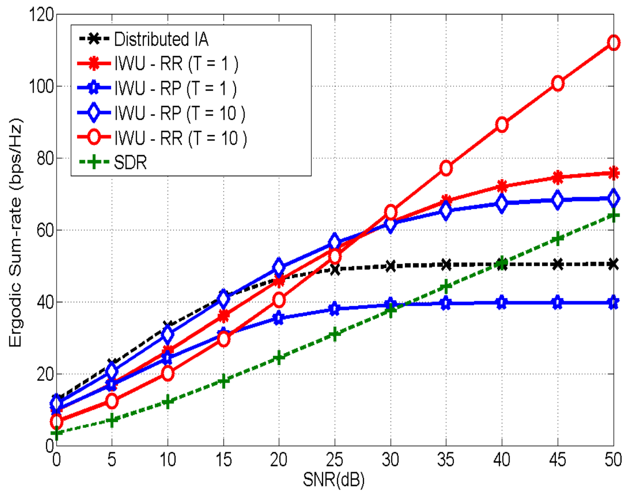


Fig. 3: Sum-rate of proposed schemes for $10 \times 10$ MIMO IC, 4 users ($d = 4$, $L = 2$), for different number of turbo iterations

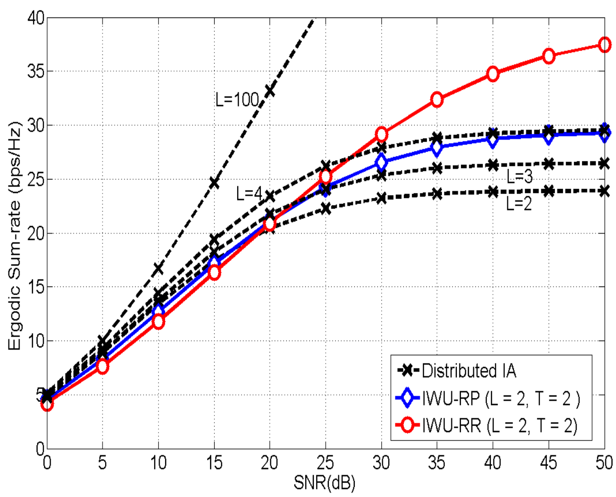


Fig. 4: Sum-rate of proposed schemes for a $4 \times 4$ MIMO IC, 3 users ($d = 2$, $L = 2$), v/s distributed IA for different number of F-B iterations

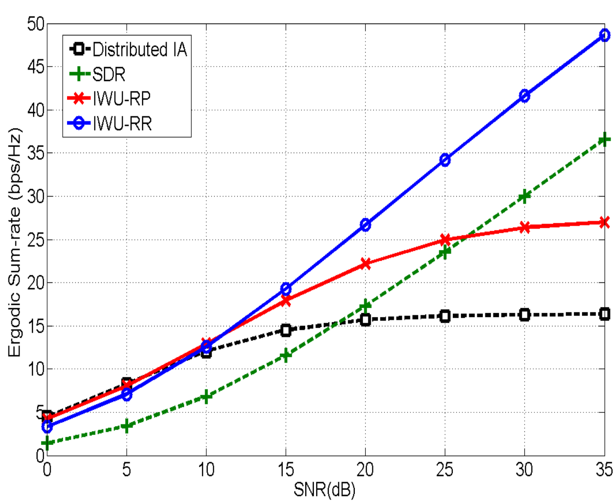


Fig. 5: Sum-rate performance in more realistic setting ($8 \times 8$ MIMO IC, 4-users, $d = 4$, $L = 2$, $T = 4$)

observe that the high-SNR slope for IWU-RR ($T = 10$) is higher than that of SDR, implying that on average, IWU-RR yields transmit / receive filters whose rank is larger than 1.

Moreover, we observe from Fig. 3 that the performance gap is very pronounced, e.g., the high-SNR spectral efficiency of IWU-RR with 10 turbo iterations is almost double that of distributed IA. Interestingly, note that for $T = 10$ IWU-RR can achieve, albeit not optimal, some degrees-of-freedom gain (shown by the linear scaling of the sum-rate at high SNR), with just 2 F-B iterations. The latter does strongly suggest that *the gains of the current approach become more accentuated, as the dimensions of the system grow*.

**Remark 2.** One might be led to think at this point that the impressive gain in sum-rate for the proposed schemes comes from the fact that the rank reduction transforms the initial IA problem into one of smaller dimensions (while distributed IA is solving the original problem), and thus that the latter simulations do not provide a basis for a fair comparison. However, this argument can be directly refuted by comparing the sum-rate performance of distributed IA, with the rank-preserving scheme (IWU-RP): as seen in Fig. 3, although both schemes yield full-rank precoders, IWU-RP still significantly outperforms distributed IA (the gap also increases with the number of turbo iterations, and as the dimensions of the problem grow). This seems to suggest that those gains follow from introducing the turbo iteration (for both schemes), and additionally from solving a relaxed problem (in the case of IWU-RR).

Next, we fix both the number of F-B and turbo iterations in our schemes to 2 and simulate the performance of distributed IA for a varying number of F-B iterations $L$ (for a feasible $4 \times 4$ MIMO IC, with $d = 2$). Fig. 4 clearly shows that for $L = 2$ and $L = 3$, distributed IA has a similar performance as both our schemes in the medium-to-low SNR region (and a worse one in the high-SNR region). It is only for $L = 4$ that it starts to outperform them in the medium-to-low SNR region only. This implies that *the overhead requirement of distributed IA is at least* 50% *more than our schemes*, for this particular case (further simulations suggest that this trend increases with the system dimensions). Moreover, we see that distributed IA delivers its “optimal” performance after a large enough number of F-B are run (corresponding to extremely high communication overhead): this suggests that the poor performance of dist IA in all simulations is due to the fact that there is significant interference leakage for small values of $L$.

## *C. Performance in more realistic setup*

In view of having a more realistic assessment - albeit still far from accurate - of the algorithms’ performance, in somewhat more practical environments, we simulate $8 \times 8$ MIMO transmission with 4 cells, 1 user per cell, 4 streams per user (fixing $L = 2$ for all algorithms, and $T = 4$ for our algorithms). We modify the cross-channel gains, $\{\boldsymbol{H}^{[kj]} |\ k \neq j\}$, such that the resulting SIR is $-5$dB, to (coarsely) emulate cell edge users. We can see from Fig. 5 that though both schemes have a similar performance as distributed IA in the very low-SNR region, they outperform it for SNR values greater than 7dB (the gap being increasing with the SNR): *IWU-RR outperforms distributed IA by* $\sim 30\%$ *at* $15$*dB of SNR, and* $\sim 80\%$ *at* $20$*dB*. This indeed shows that our schemes are good candidates for operating in such practical scenarios. On another note, we also see that IWU-RR and SDR have a similar high-SNR slope (thus implying that IWU-RR finds a rank-one solution in almost all cases). However, the massive gap between IWU-RR and SDR, indicates that the solution provided by the IWU-RR yields significantly higher effective channel gain than the solution found by the SDR.

### D. Discussions

It is interesting to notice in Fig. 3-5 that the gains for both schemes seem to happen in the medium-to-high SNR region: this is expected, since in that regime, reducing interference is vital to increasing the sum-rate. The observed performance boosts for both IWU-RR and IWU-RP are attributed to the introduction of the turbo-iteration. Furthermore, in the case of IWU-RR, the massive performance gain additionally comes from the fact it is solving a relaxed problem. On another note, Fig. 3 shows that indeed the optimal rank-one solution to (4) provided by SDR is massively suboptimal in terms of sum-rate performance. This also provides a clear motivation for our work, where the proposed algorithms were mainly aimed at avoiding this rank-one solution.

Though negligible, one can indeed see a degradation in performance of both schemes, with respect to distributed IA, in the low-SNR region (as seen from Fig. 3-6). Despite the that fact that full-rank filters are known to be optimal in the high-SNR regime (thanks to the insights from interference alignment), in the very low-SNR (interference-free) regime however, matched filtering is the optimal strategy, and consequently rank-one filters are optimal as well. We note that although our rank-reducing algorithm does find a rank-one solution, it might be the "wrong one", i.e., different from the matched filtering direction: this is due to the fact that both our algorithm and distributed IA look for solutions that reduce interference, that most likely are not aligned with the matched filtering direction. On the other hand, the full-rank solution given by distributed IA is likely to transmit a reasonable amount of energy along that direction. This might explain the reason that distributed IA exhibits better performance than IWU-RR, in the low-SNR region. Moreover, recall that schemes such as the proposed ones and distributed IA do not take into account the desired signal and noise subspace. As a result, one can at best speculate about their low SNR behavior (since the SNR is not part of their mathematical formulation). However, referring to Fig. 6, we can see that this degradation is minimal (around $5\%$ for IWU-RP and $8\%$ for IWU-RR, over the benchmark scheme). A possible alternative to mitigate this issue is to select the scheme based on the operating SNR, i.e., select IWU-RP in the low-SNR region, since it has a similar performance as distributed IA (as seen from Fig. 3-6): this can be easily implemented since both algorithms have the exact same structure, and only the updates have to be changed.

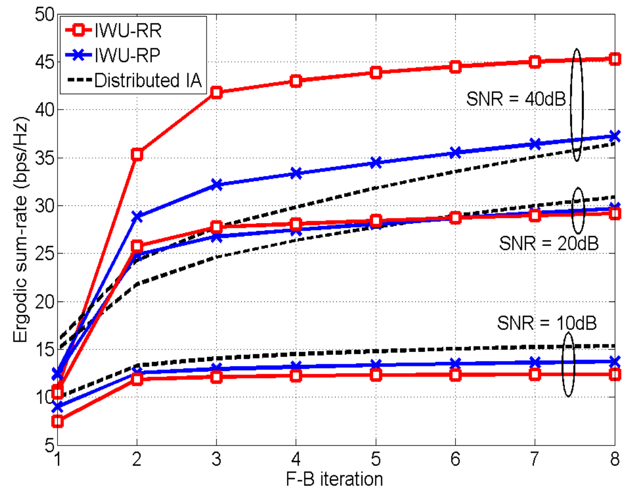


Fig. 6: Ergodic sum-rate of proposed schemes vs distributed IA, as a function of operating SNR ($4 \times 4$, 3-user MIMO IC $d = 2$, $L = 2$)

In conclusion, though both schemes are extremely similar in their algorithmic structure (i.e. both update the filter weights within a turbo iteration), both are distributed, optimize the same metric, and require the same (local) CSI quantities at each node, they indeed have some fundamental differences. The fact that the filter update equations are different has several implications: the update IWU-RR does not necessarily lead to full rank filters, and though it was shown that IWU-RR attempts to solve the relaxed problem in (4), such claim cannot be made for IWU-RP mainly due to the different constraints on the update structure. Finally, we compared their performance in several scenarios via simulations, and suggested reasons for the behavior we observed.

## VII. Conclusion

Within the context of the leakage minimization problem, we proposed two distinct schemes based on rank-reducing (IWU-RR) and rank-preserving (IWU-RP) filter updates, where the transmit and receive filter weights are iteratively refined in a turbo-like structure. We then showed that they are well suited for delivering high spectral efficiency (compared to the well-known distributed IA algorithm), while generating very small overhead (typically, only a few F-B iterations). Though the introduction of the so-called turbo iteration significantly boosted the performance of both schemes, it is clear that its impact was much more significant when combined with the rank-reducing updates in IWU-RR, thus allowing it to achieve a performance that otherwise required a much larger number of F-B iterations. In that sense, the proposed schemes enabled us to tradeoff the communication overhead associated with the F-B iterations - a rather expensive resource, with computational complexity (an immensely cheaper resource).

## APPENDIX

### A. Proof of Proposition 1

Given $\boldsymbol{U}$ and $\boldsymbol{Q}$, and using the fact that $\boldsymbol{Q}$ and $\boldsymbol{Q}^\perp$ are unitary and orthogonal, the proof is simple after noting that any subspace $\boldsymbol{U}$ can be expressed as a sum of its components over orthogonal directions (a result that trivially follows from the orthogonal decomposition theorem), i.e. $\boldsymbol{U} = \boldsymbol{P}\boldsymbol{U} + \boldsymbol{P}^\perp\boldsymbol{U}$, where $\boldsymbol{P}$ and $\boldsymbol{P}^\perp$ are any two orthogonal projection matrices. In particular, let $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{Q}^\dagger$ and $\boldsymbol{P}^\perp = \boldsymbol{Q}^\perp{\boldsymbol{Q}^\perp}^\dagger$, then $\boldsymbol{U} = \boldsymbol{Q}\boldsymbol{Q}^\dagger\boldsymbol{U} + \boldsymbol{Q}^\perp{\boldsymbol{Q}^\perp}^\dagger\boldsymbol{U} = \boldsymbol{Q}\boldsymbol{A} + \boldsymbol{Q}^\perp\boldsymbol{B}$, where $\boldsymbol{A} = \boldsymbol{Q}^\dagger\boldsymbol{U}$ and $\boldsymbol{B} = {\boldsymbol{Q}^\perp}^\dagger\boldsymbol{U}$

### B. Proof of Lemma 1

The result is a special case of [23], which shows that strong duality holds for all complex valued quadratic problems with up to two quadratic inequality constraints. It is straightforward to show that (8) and its dual are strictly feasible. Furthermore, since the equality constraint $\|\boldsymbol{X}\|_F^2 = \zeta$ is equivalent to the two inequality constraints $\|\boldsymbol{X}\|_F^2 \le \zeta$ and $\|\boldsymbol{X}\|_F^2 \ge \zeta$, the results of [23] show that the globally optimum solution of (8) can be obtained from its dual. For the specific formulation (8), the solution takes a particularly simple form. Adding the Lagrange multipliers of the two inequality constraints into a single dual variable $\mu$, the necessary and sufficient conditions of [23, Theorem 2.4] can be written as

$$\left(\gamma_2^2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T} + \mu^\star\boldsymbol{I}\right)\boldsymbol{X}^\star = -\gamma_1\gamma_2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{\Theta} \tag{26}$$

$$\|\boldsymbol{X}\|_F^2 = \zeta \tag{27}$$

$$\left(\gamma_2^2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T} + \mu^\star\boldsymbol{I}\right) \succeq 0\ . \tag{28}$$

The last inequality is fulfilled when $\mu > -\gamma_2^2\ \lambda_1[\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T}]$ ($\mu = -\gamma_2^2\ \lambda_1[\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T}]$ can be excluded since it results in $\|\boldsymbol{X}\|_F^2 = \infty$). Next, we study $g(\mu) \stackrel{\triangle}{=} \|\boldsymbol{X}^\star(\mu)\|_F^2 - \zeta$. Let $\sigma_1, ..., \sigma_d$ be the eigenvalues of $\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T}$ (sorted in increasing order), and $\boldsymbol{v}_1, ..., \boldsymbol{v}_d$ their corresponding eigenvectors. We first rewrite $g(\mu)$ as

$$\begin{aligned} g(\mu) &= \gamma_1^2\gamma_2^2\ \mathrm{tr}[\ \boldsymbol{\Theta}^\dagger\boldsymbol{Q}\boldsymbol{T}(\gamma_2^2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T} + \mu\boldsymbol{I})^{-2}\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{\Theta}\ ] - \zeta \\ &= \gamma_1^2\gamma_2^2\ \mathrm{tr}[\boldsymbol{X}_o^\dagger(\ \gamma_2^2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T} + \mu\boldsymbol{I}\ )^{-2}\boldsymbol{X}_o] - \zeta\ , \end{aligned}$$

where $\boldsymbol{X}_o = \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{\Theta}$. Note that we can express the matrix $(\gamma_2^2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T} + \mu\boldsymbol{I})^{-2}$ as a function of $\sigma_i, \boldsymbol{v}_i, \mu$, as $(\gamma_2^2\ \boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T} + \mu\boldsymbol{I})^{-2} = \sum_{i=1}^d(\gamma_2^2\sigma_i + \mu)^{-2}\boldsymbol{v}_i\boldsymbol{v}_i^\dagger$. Thus, we rewrite $g(\mu)$ as follows,

$$\begin{aligned} g(\mu) &= \gamma_1^2\gamma_2^2\ \mathrm{tr}[\boldsymbol{X}_o^\dagger(\sum_{i=1}^d(\gamma_2^2\sigma_i + \mu)^{-2}\boldsymbol{v}_i\boldsymbol{v}_i^\dagger)\boldsymbol{X}_o] - \zeta \\ &= \sum_{i=1}^d \frac{\gamma_1^2\gamma_2^2\ \mathrm{tr}(\boldsymbol{X}_o^\dagger\boldsymbol{v}_i\boldsymbol{v}_i^\dagger\boldsymbol{X}_o)}{(\gamma_2^2\sigma_i + \mu)^2} - \zeta = \sum_{i=1}^d \frac{(\gamma_1\gamma_2\ c_i)^2}{(\gamma_2^2\sigma_i + \mu)^2} - \zeta, \end{aligned} \tag{29}$$

where $c_i = \|\boldsymbol{X}_o^\dagger\boldsymbol{v}_i\|_2$. A quick look at this last expression reveals that indeed $g(\mu)$ is strictly monotonically decreasing in $\mu$, for $\mu > -\gamma_2^2\sigma_1 = -\gamma_2^2\lambda_1[\boldsymbol{T}^\dagger\boldsymbol{Q}\boldsymbol{T}]$. Consequently, $g(\mu) = 0$ has a unique solution. To find the upper bound on $\mu$ to use in a bisection search, note that if $\mu > 0$ then

$$\begin{aligned} g(\mu) &< \gamma_1^2\gamma_2^2\sum_{i=1}^d \frac{\|\boldsymbol{X}_o^\dagger\boldsymbol{v}_i\|^2}{\mu^2} - \zeta = \frac{\gamma_1^2\gamma_2^2}{\mu^2}\|\boldsymbol{X}_o^\dagger\boldsymbol{V}_o\|_F^2 - \zeta \\ &= \left(\frac{\gamma_1\gamma_2\|\boldsymbol{X}_o^\dagger\|_F}{\mu}\right)^2 - \zeta\ . \end{aligned}$$

where $\boldsymbol{V}_o = [\boldsymbol{v}_1\ \ \ldots\ \ \boldsymbol{v}_d]$. Consequently if $\mu \ge \gamma_1\gamma_2\|\boldsymbol{X}_o^\dagger\|_F/\sqrt{\zeta}$, we get $g(\mu) < 0$. This concludes the proof.

### C. Proof of Lemma 2

Let $\mathcal{S}_k$ be the set of local and global minima of $(K2)$, which can written as,

$$\mathcal{S}_k = \{\ x \mid p'(x) = 0,\ p''(x) \ge 0,\quad 0 \le x < 1\},$$

where $p(x) = (1-x^2)e_1 + x\sqrt{1-x^2}e_2 + x^2e_3$. We will show that the above set has a single element, thereby establishing that $(K2)$ is a convex problem, and derive the solution. Defining $a = e_1 - e_3$, we start by finding the zero-differential points of $p(x)$, i.e., $p'(x) = 0 \Rightarrow e_2\frac{1-2x^2}{\sqrt{1-x^2}} = 2ax \Rightarrow 4(a^2 + e_2^2)x^4 - 4(a^2 + e_2^2)x^2 + e_2^2 = 0\ (e.1)$ where the last equation stems from squaring both sides. Note that some of the roots of $(e.1)$ will not correspond to zero-differential points (we will remedy this fact later). Letting $X = x^2$, we can rewrite $(e.1)$ and its solution as,

$$\begin{aligned} &4(a^2 + e_2^2)X^2 - 4(a^2 + e_2^2)X + e_2^2 = 0,\ \ (e.2) \\ &X_1 = 1/2 + a/2\sqrt{a^2 + e_2^2},\quad X_2 = 1/2 - a/2\sqrt{a^2 + e_2^2}. \end{aligned} \tag{30}$$

Moreover, since we are interested in solutions to $(K2)$ that lie in the interval $[0,\ 1]$, we verify that indeed $X_1, X_2$ lie in this interval. This can be easily done by considering two cases, $a \ge 0$ and $a \le 0$. For the former, we write,

$$\begin{aligned} &0 \le a^2 \le a^2 + e_2^2 \Rightarrow 0 \le a \le \sqrt{a^2 + e_2^2} \Rightarrow 0 \le a/\sqrt{a^2 + e_2^2} \le 1, \\ &\quad\Rightarrow 1 \le 1 + a/\sqrt{a^2 + e_2^2} \le 2 \Rightarrow 1/2 \le X_1 \le 1. \\ &\quad\Rightarrow 0 \le 1 - a/\sqrt{a^2 + e_2^2} \le 1 \Rightarrow 0 \le X_2 \le 1/2. \end{aligned} \tag{31}$$

Using exactly the same manner, we can show that if $a \le 0$, then $0 \le X_1 \le 1/2$ and $1/2 \le X_2 \le 1$, thus concluding that both lie in the interval $[0,\ 1]$. This said, by discarding negative solutions, the solution to $(e.1)$ is $x_1 = \sqrt{X_1},\ x_2 = \sqrt{X_2}$, i.e.,

$$x_1 = \sqrt{X_1} = \sqrt{1/2 + a/2\sqrt{a^2 + e_2^2}},$$

$$x_2 = \sqrt{X_2} = \sqrt{1/2 - a/2\sqrt{a^2 + e_2^2}}.$$

Note that both $x_1$ and $x_2$, lie in the interval $[0, 1]$. Recall that not all the solutions of $(e.1)$ correspond to zero-differential points of $p(x)$ - in fact it is easy to show that $p'(x_1) = 0$ and $p'(x_2) \ne 0$, implying that $p(x)$ has a single unique zero-differential point at $x_1$. Thus, it remains to show that $p''(x_1) = 0$. Using the fact that $x_1^2 = X_1$, $x_2^2 = X_2$, and noting that $X_1 + X_2 = 1$, we rewrite this condition as,

$$p''(x_1) \ge 0 \Leftrightarrow -2a - e_2\left[\left(\frac{X_1}{X_2}\right)^{3/2} + 3\left(\frac{X_1}{X_2}\right)^{1/2}\right] \ge 0$$

The last equation can be easily shown, by plugging in the values for $X_1$ and $X_2$ (we will omit the derivations since they are rather straightforward and easily reproduced).

Thus, we conclude that the set of global minima of $p(x)$, $\mathcal{S}_k$, has a single element, thereby establishing that $p(x)$ is convex and has single global minimum given by $\mathcal{S}_k = \{\sqrt{1/2 + a/2\sqrt{a^2+e_2^2}}\}$.

## ACKNOWLEDGEMENT

The first author wishes to acknowledge the insightful comments of Mr. Rasmus Brandt.

## REFERENCES

[1] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 9, pp. 1380–1408, 2010.

[2] V. Cadambe and S. Jafar, "Interference alignment and degrees of freedom of the K -user interference channel," *Information Theory, IEEE Transactions on*, vol. 54, no. 8, pp. 3425–3441, 2008.

[3] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Transactions on Information Theory*, vol. 57, pp. 3309–3322, June 2011.

[4] H. Ghauch and C. Papadias, "Interference alignment: A one-sided approach," in *2011 IEEE Global Communications Conference (GLOBECOM 2011)*, pp. 1 –5, Dec. 2011.

[5] D. Schmidt, C. Shi, R. Berry, M. Honig, and W. Utschick, "Minimum mean squared error interference alignment," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 1106 –1110, Nov. 2009.

[6] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[7] S. W. Peters and R. W. Heath, "Cooperative algorithms for MIMO interference channels," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 206–218, Jan. 2011.

[8] I. Santamaria, O. Gonzalez, R. W. Heath, and S. W. Peters, "Maximum sum-rate interference alignment algorithms for MIMO channels," in *2010 IEEE Global Communications Conference (GLOBECOM 2010)*, pp. 1–6, IEEE, Dec. 2010.

[9] D. Schmidt, C. Shi, R. Berry, M. Honig, and W. Utschick, "Comparison of distributed beamforming algorithms for MIMO interference networks," *Signal Processing, IEEE Transactions on*, vol. 61, pp. 3476–3489, July 2013.

[10] C. M. Yetis, T. Gou, S. A. Jafar, and A. H. Kayran, "On feasibility of interference alignment in MIMO interference networks," *IEEE Transactions on Signal Processing*, vol. 58, pp. 4771–4782, Sept. 2010.

[11] H. Bolcskei and I. J. Thukral, "Interference alignment with limited feedback," in *IEEE International Symposium on Information Theory, 2009. ISIT 2009*, pp. 1759–1763, IEEE, July 2009.

[12] R. Krishnamachari and M. Varanasi, "Interference alignment under limited feedback for MIMO interference channels," in *2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 619 –623, June 2010.

[13] J.-H. Chang, L. Tassiulas, and F. Rashid-Farrokhi, "Joint transmitter receiver diversity for efficient space division multiaccess," *Wireless Communications, IEEE Transactions on*, vol. 1, pp. 16–27, Jan 2002.

[14] M. Bengtsson, "A pragmatic approach to multi-user spatial multiplexing," in *Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2002*, pp. 130–134, Aug 2002.

[15] O. El Ayach, A. Lozano, and R. Heath, "On the overhead of interference alignment: Training, feedback, and cooperation," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 11, pp. 4192–4203, 2012.

[16] O. El Ayach, A. Lozano, and R. Heath, "Optimizing training and feedback for MIMO interference alignment," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 1717–1721, 2011.

[17] P. Mohapatra, K. E. Nissar, and C. Murthy, "Interference alignment algorithms for the k user constant MIMO interference channel," *Signal Processing, IEEE Transactions on*, vol. 59, pp. 5499–5508, Nov 2011.

[18] Z.-Q. Luo, W.-K. Ma, A.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *Signal Processing Magazine, IEEE*, vol. 27, pp. 20–34, May 2010.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[20] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A Unified Convergence Analysis of Block Successive Minimization Methods for Nonsmooth Optimization," *SIAM Journal on Optimization*, vol. 23, sep 2012.

[21] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[22] M. Razaviyayn, G. Lyubeznik, and Z.-Q. Luo, "On the degrees of freedom achievable through interference alignment in a MIMO interference channel," *Signal Processing, IEEE Transactions on*, vol. 60, pp. 812–821, Feb 2012.

[23] A. Beck and Y. Eldar, "Strong duality in nonconvex quadratic optimization with two quadratic constraints," *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 844–860, 2006.