

# Interference Alignment: A one-sided approach

Hadi G. Ghauch

Information Networking Institute  
Carnegie Mellon University  
Pittsburgh, PA, USA  
hghauch@cmu.edu

Constantinos B. Papadias

Broadband Wireless & Network Sensors Lab  
Athens Information Technology  
Athens, Greece  
papadias@ait.gr

**Abstract**—Interference Alignment (IA) is the process of designing signals in such a way that they cast overlapping shadows at their unintended receivers, while remaining distinguishable at the intended ones [1]. Our goal in this paper is to come up with an algorithm for IA that runs at the transmitters only (and is transparent to the receivers), that doesn't require channel reciprocity, and thus alleviates the need to alternate between the forward and reverse network as is the case in [2], thereby saving significant overhead in certain environments where the channel changes frequently. Most importantly, our effort is focused on ensuring that this one-sided approach does not degrade the performance of the system w.r.t. [2]. As a first step, we mathematically express the interference in each receiver's desired signal as a function of the transmitters' beamforming vectors. We then propose a simple steepest descent (SD) algorithm and use it to minimize the interference in each receiver's desired signal space. We mathematically establish equivalences between our approach and the Distributed IA algorithm presented in [2] and show that our algorithm also converges to an alignment solution (when the solution is feasible).

**Keywords:** *Interference Alignment, Interference Channel, Matrix Differentials/Derivatives, Optimization, Steepest Descent.*

## I. INTRODUCTION

Interference Alignment (IA) is a recent technique to achieve the sum-rate (SR) degrees of freedom (DoF) of the  $K$ -user  $M \times M$  MIMO Interference Channel (IC),

$$C_{sr} = (KM/2) \log(1+P) + o(\log(P)) \quad (1)$$

(first presented implicitly in [4] and then further elaborated in [1], [5]). The key to the achievability of this result is forcing each transmitter to use only half of its signaling space, and each receiver to partition its received signal space into two equally sized subspaces: one is intended for the desired signal, while the other is left for all the interference [3]. As a result, each transmitter-receiver pair can "talk" over an interference-free space, irrespective of the number of interferers.

In a nutshell, the problem of IA boils down to finding transmit precoding ( $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}$ ) and receive interference suppression matrices ( $\mathbf{U}^{[1]}, \dots, \mathbf{U}^{[K]}$ ), to cancel all the unwanted interference at each receiver. An elegant iterative algorithm that exploits channel reciprocity to alternate between the forward and reverse network, and find such matrices in a distributed way, was presented in [2]. However, such an approach exhibits several aspects that might be perceived as drawbacks if one wishes to apply this algorithm to certain environments. For instance, alternating between the forward and reverse network requires tight synchronization at

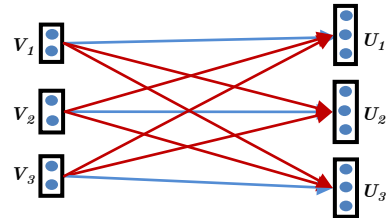


Figure 1: 3-user 2x3 MIMO IC

both ends (which may be hard to achieve, especially at the receivers' side). Moreover, this alternation might induce significant overhead in a dynamic environment where the channel varies rapidly. Furthermore, the receivers, having generally limited computational complexity, might be a bottleneck for the execution time of the algorithm. And finally, the assumption of channel reciprocity practically limits the applicability of such an algorithm to TDD systems.

As is the case in [2], most of the IA algorithms presented in the literature such as [6] and [7], perform the optimization of some cost function over both the precoding matrices and interference suppression filters i.e.  $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}, \mathbf{U}^{[1]}, \dots, \mathbf{U}^{[K]}$ . Although such an algorithm is desirable in terms of performance, an inevitable consequence is the fact that both the transmitters and receivers are active in the algorithm, an implication that we may wish to avoid for the reasons that we previously stated. Our aim in this paper is to present an approach that decouples the IA problem, by restricting the optimization to the transmitters' side only. A clear advantage of such an approach is the fact that, as seen by the receivers, the algorithm is *transparent* (g1), i.e. all a receiver has to do is to pick the subspace with the lowest interference (not more than what is already in use). Moreover, it *bypasses the overhead* and other complications generated by alternating between the forward and reverse network (g2). Furthermore, by adopting this approach, we *relax the assumption of channel reciprocity* and make our algorithm applicable to both TDD and FDD systems (g3). We keep in mind that we do not expect our method to outperform the results of [2], and especially other non-subspace methods [7], at low interference-to-noise ratio (INR). In fact, by reducing the search space from  $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}, \mathbf{U}^{[1]}, \dots, \mathbf{U}^{[K]}$  to  $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}$  only, it is unlikely that we end up with a better solution. However, from the start, our aim was to design a simple algorithm with the above guidelines in mind (g1-g3). This said, the bulk of the effort is to ensure that the approach is feasible and does not incur a significant loss in performance. In the following, bold uppercase letters denote matrices / vectors. We assume that

the eigenvalues of a matrix  $\mathbf{A}$ , and their corresponding eigenvectors, are sorted in increasing order. Therefore,  $\lambda_i[\mathbf{A}]$  denotes the  $i^{\text{th}}$  eigenvalue of  $\mathbf{A}$ . The  $^H$  symbol denotes the conjugate transpose of a matrix (Hermitian), while the operator  $\langle \mathbf{Z}, \mathbf{Z} \rangle$  denotes the inner product of a vector/matrix  $\mathbf{Z}$ . Moreover, we denote by  $\mathcal{K} = \{1, \dots, K\}$  the set of integers from 1 to  $K$ . Finally,  $\mathbf{I}_n$  denotes the  $n \times n$  identity matrix.

## II. SIGNAL MODEL

We build upon the model introduced in [2]. We consider a  $K$ -user MIMO Interference Channel (IC) where the  $k^{\text{th}}$  transmitter-receiver pair is equipped with  $M^{[k]}$  and  $N^{[k]}$  antennas, respectively.  $d^{[k]}$  is the desired number of streams between the  $k^{\text{th}}$  transmitter-receiver pair, where  $d^{[k]} \leq \min(M^{[k]}, N^{[k]})$ . Moreover,  $\mathbf{H}^{[kj]}$  denotes the  $N^{[k]} \times M^{[j]}$  channel matrix from transmitter  $j$  to receiver  $k$ , and is assumed to have i.i.d complex Gaussian random variables, drawn from a continuous distribution. Finally,  $\mathbf{V}^{[j]}$  denotes the  $j^{\text{th}}$  transmitter's  $M^{[j]} \times d^{[j]}$  precoding matrix ( $1 \leq j \leq K$ ), whose orthonormal columns span the  $d$ -dimensional space at the transmitter. We denote the received signal vector at receiver  $k$  after interference suppression by

$$\mathbf{Y}_s^{[k]} = \mathbf{U}^{[k]H} \mathbf{Y}^{[k]} = \mathbf{U}^{[k]H} \left( \sum_{j=1}^K \mathbf{H}^{[kj]} \mathbf{V}^{[j]} \mathbf{X}^{[j]} + \mathbf{Z}^{[k]} \right), k \in \mathcal{K} \quad (2)$$

where  $\mathbf{U}^{[k]}$  is the  $N^{[k]} \times d^{[k]}$  interference suppression filter at receiver  $k$ ,  $\mathbf{X}^{[j]}$  is the  $d^{[j]} \times 1$  vector of independently encoded Gaussian symbols of transmitter  $j$ , with covariance matrix  $(P^{[j]}/d^{[j]})\mathbf{I}_{d^{[j]}}$ , and  $\mathbf{Z}^{[k]}$  is the i.i.d complex Gaussian noise at receiver  $k$  with unit variance.

## III. ALGORITHM DERIVATION

### A. Motivation

In the following, we assume for simplicity that  $(M^{[k]}, N^{[k]}, d^{[k]}) = (M, N, d), \forall k \in \mathcal{K}$ . As we previously stated, the IA conditions imply that we need to make the interference at each receiver align by occupying an  $(N-d)$ -dimensional subspace, creating an interference-free,  $d$ -dimensional space for the desired signal (where  $d$  must equal  $N/2$  to achieve the MIMO IC degrees of freedom (DoF), i.e.  $KM/2$ , only feasible when  $K \leq 3$ ). Equivalently, if we are able to create at each receiver a  $d$ -dimensional signal space that is free from interference, we have implicitly aligned the interference in the other remaining space. This done, all a receiver has to do is to “hide from the interference” by projecting the received signal onto the  $d$ -dimensional subspace that has the lowest interference, thereby effectively suppressing all the undesired interference (Fig. 2).

In that sense, our algorithm should aim at creating a  $d$ -dimensional subspace that is free from interference, at each receiver. The natural question that arises is: what is a good metric (cost function) for such a purpose? We know from Principal / Minor Component Analysis that the eigenvectors of the interference covariance matrix correspond to the

dimensions along which a receiver “sees” interference, and their corresponding eigenvalues indicate the variance / power of interference along that dimension (Fig. 2). Assuming that the eigenvalues and their

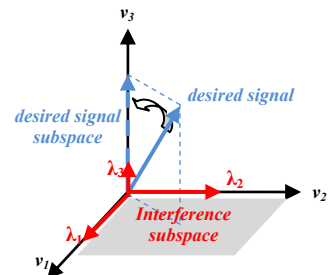


Figure 2: 3D signal space at Rx corresponding eigenvectors are sorted in increasing order, this implies that the  $d$ -dimensional subspace with the lowest interference is spanned by the  $d$ -eigenvectors corresponding to the  $d$ -smallest eigenvalues, and the variance of interference in this subspace is given by the sum of the  $d$ -smallest eigenvalues (used in [2]). Summing up, we wish to track the variance of interference in the desired signal space, by tracking the sum of the  $d$ -smallest eigenvalues of the interference covariance matrix at each receiver, as the transmitters vary their precoding matrices.

### B. Problem formulation

This said, at each receiver, we seek to minimize the sum of the  $d$ -smallest eigenvalues of the interference covariance matrix, over the set of transmit precoding matrices  $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}$ . Thus, we define our cost function:

$$\min_{\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}} f = \sum_{k=1}^K \sum_{i=1}^{d^{[k]}} \lambda_i[\mathbf{Q}^{[k]}], k \in \mathcal{K} \quad (3)$$

subject to  $\mathbf{V}^{[j]H} \mathbf{V}^{[j]} = \mathbf{I}_d, \forall j \in \mathcal{K}$

$\mathbf{Q}^{[k]}$  is the  $k^{\text{th}}$  receiver's interference covariance matrix:

$$\mathbf{Q}^{[k]} = \sum_{j=1, j \neq k}^K \frac{P^{[j]}}{d^{[j]}} \mathbf{H}^{[kj]} \mathbf{V}^{[j]} \mathbf{V}^{[j]H} \mathbf{H}^{[kj]H} \quad (4)$$

Now that we defined the cost function, we need a way to minimize it. Intuitively, one first thinks about derivatives. However, in this case, the variables in question are matrices (we need derivatives of the form  $d\mathbf{Q}/d\mathbf{V}$ , where both  $\mathbf{Q}$  and  $\mathbf{V}$  are matrices).

### C. Mathematical Model

Although our goal is to propose an algorithm for IA, another notable feature of this work is the novel mathematical expression that we derive to account for interference in the network. It draws its foundations in matrix differential calculus (matrix differentials [10]). By using matrix differentials (which can be thought of as derivatives for matrices), we hope to find expressions of the form  $df = \mathbf{DF} \cdot d\mathbf{V}$  where  $df$  and  $d\mathbf{V}$  are the differential of the function and of the variable, respectively, and where  $\mathbf{DF}$  is a Jacobian matrix (a matrix of partial derivatives) that translates infinitesimal changes in  $\mathbf{V}$  (i.e.  $d\mathbf{V}$ ), to changes in  $f$  [10]. The reason for seeking such matrices is the fact that Jacobian matrices go hand-in-hand with numerical minimization. Due to space limitations, we only present the results (the full

derivations are shown in [15], Part II, section C). Two blocks are chained together to yield the final expression.

### 1) Differentials of Covariance Matrices

We show mathematically that for a  $K$ -user MIMO IC, the interference covariance matrix at receiver  $k$  satisfies:

$$\text{vec}(d\mathbf{Q}^{[k]}) = \sum_{j=1, j \neq k}^K \mathbf{D}\mathbf{Q}_R^{[kj]} \text{vec}(d\mathbf{V}_R^{[j]}) + \mathbf{D}\mathbf{Q}_I^{[kj]} \text{vec}(d\mathbf{V}_I^{[j]}), \forall k \in \mathcal{K} \quad (5)$$

where the  $\text{vec}()$  operator stacks the columns of the operand matrix vertically;  $\mathbf{V}_R$  and  $\mathbf{V}_I$  denote the real and imaginary parts of  $\mathbf{V}$ , respectively. The above equation answers the following question: *How do the entries in  $\mathbf{Q}^{[k]}$  vary ( $1 \leq k \leq K$ ), when any element in  $\mathbf{V}^{[j]}$  changes ( $1 \leq j \leq K$ )?* In other words, the above equation relates the changes in any transmitter's signaling vectors, to all the changes that they induce at the corresponding covariance matrices.

### 2) Differentials of Eigenvalues of Covariance Matrices

Since we are ultimately interested in minimizing sums of eigenvalues (3), the second logical step would be to derive an expression that relates, through a Jacobian matrix  $\mathbf{D}\mathbf{L}$ , the differential of a sum of eigenvalues of a given covariance matrix,  $d\lambda_s$ , to the differential of the covariance matrix itself  $\mathbf{D}\mathbf{Q}$ , (again, an expression of the form  $d\lambda = \mathbf{D}\mathbf{L}\mathbf{D}\mathbf{Q}$ ). We extend a result presented in [9], and derive the following expression. For a given covariance matrix  $\mathbf{Q}^{[kj]}$  (of real eigenvalues, since  $\mathbf{Q}^{[kj]}$  is Hermitian), the differential of the sum of its  $d$ -smallest eigenvalues can be written as:

$$d(\lambda_s(\mathbf{Q}^{[kj]})) = \left( \sum_{i=1}^{d^{[kj]}} \mathbf{D}\mathbf{L}_i^{[kj]} \right) \text{vec}(d\mathbf{Q}^{[kj]}) = \mathbf{D}\mathbf{L}_s^{[kj]} \text{vec}(d\mathbf{Q}^{[kj]}), \forall k \in \mathcal{K} \quad (6)$$

$$\text{where } \lambda_s(\mathbf{Q}^{[kj]}) = \sum_{i=1}^{d^{[kj]}} \lambda_i(\mathbf{Q}^{[kj]})$$

represents the sum of the  $d$ -smallest eigenvalues of a covariance matrix  $\mathbf{Q}^{[kj]}$ , and  $\mathbf{D}\mathbf{L}_i^{[kj]}$  is the Jacobian matrix associated with the  $i^{\text{th}}$  eigenvalue of  $\mathbf{Q}^{[kj]}$  (here we also omit the expression for  $\mathbf{D}\mathbf{L}_i^{[kj]}$ ).

### 3) Chaining the two blocks

To combine the two blocks, we substitute the expression of  $\text{vec}(d\mathbf{Q}^{[kj]})$  in (5) into (6), to get:

$$d(\lambda_s(\mathbf{Q}^{[k]})) = \mathbf{D}\mathbf{L}_s^{[k]} \sum_{j=1, j \neq k}^K \mathbf{D}\mathbf{Q}_R^{[kj]} \text{vec}(d\mathbf{V}_R^{[j]}) + \mathbf{D}\mathbf{Q}_I^{[kj]} \text{vec}(d\mathbf{V}_I^{[j]}), k \in \mathcal{K} \quad (7)$$

where  $k \in \mathcal{K}$ . In this equation, we address the issue of how does the sum of the eigenvalues of the interference covariance matrix at each receiver (the variance of interference in the desired signal space) change, as any element in  $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}$  changes (the transmitters' beamforming vectors).

Using the above equation and exploiting the linearity of differentials, we write the differential of our cost function

$$df = \sum_{k=1}^K \mathbf{D}\mathbf{L}_s^{[k]} \left\{ \sum_{j=1, j \neq k}^K (\mathbf{D}\mathbf{Q}_R^{[kj]} \text{vec}(d\mathbf{V}_R^{[j]}) + \mathbf{D}\mathbf{Q}_I^{[kj]} \text{vec}(d\mathbf{V}_I^{[j]})) \right\} \quad (8)$$

We also notice that the above expression can be conveniently placed in matrix form (shown at the bottom of the page).

$$\begin{aligned} df &= \mathbf{D}\mathbf{L}^A \mathbf{D}\mathbf{Q}_R^A \text{vec}(d\mathbf{V}_R^A) + \mathbf{D}\mathbf{L}^A \mathbf{D}\mathbf{Q}_I^A \text{vec}(d\mathbf{V}_I^A) \\ &= \mathbf{D}\Psi_R^A \text{vec}(d\mathbf{V}_R^A) + \mathbf{D}\Psi_I^A \text{vec}(d\mathbf{V}_I^A) \\ &= \begin{bmatrix} \mathbf{D}\Psi_R^{[1]} & \dots & \mathbf{D}\Psi_R^{[K]} \\ & & \vdots \\ \mathbf{D}\Psi_I^{[1]} & \dots & \mathbf{D}\Psi_I^{[K]} \end{bmatrix} \begin{bmatrix} \text{vec}(d\mathbf{V}_R^{[1]}) \\ \vdots \\ \text{vec}(d\mathbf{V}_R^{[K]}) \end{bmatrix} + \begin{bmatrix} \mathbf{D}\Psi_I^{[1]} & \dots & \mathbf{D}\Psi_I^{[K]} \\ & & \vdots \\ \mathbf{D}\Psi_I^{[1]} & \dots & \mathbf{D}\Psi_I^{[K]} \end{bmatrix} \begin{bmatrix} \text{vec}(d\mathbf{V}_I^{[1]}) \\ \vdots \\ \text{vec}(d\mathbf{V}_I^{[K]}) \end{bmatrix} \quad (9) \end{aligned}$$

where in the last two equations, the  $A$  superscript denotes the augmented matrix formed by concatenating all the submatrices together. By examining the last expression for  $df$ , two statements can be made. Firstly, we have achieved our goal and expressed the differential of our cost function as  $df = \mathbf{D}\mathbf{F} \cdot d\mathbf{V}$  (actually, in our case, we have two Jacobian matrices). Thus, we have derived the Jacobian matrices that we need for performing our minimization. It is worth mentioning here that a direct analytical attempt to find the derivative of  $f$  would be tedious due to the non-linear  $\lambda_i[\dots]$  operator in our cost function. Secondly, we note that it is easy to show that (8) has the exact same form as the first order linear approximation of a scalar function of a vector  $x$ , using the gradient  $f(x) = f(x_o) + (\nabla f)_{x_o}(x - x_o)$ . Thus, effectively, we have transformed a seemingly complex optimization problem involving a set of  $K$  matrices, into a simple gradient-like expression suitable for any gradient descent algorithm.

### D. Proposed Algorithm

As previously said, our motivation for using differentials and Jacobian matrices is the fact they go hand-in-hand with numerical minimization problems. The general structure of our Steepest Descent (SD) algorithm is inspired by Algorithm 15 in [8], with the extension that every variable  $\text{vec}(d\mathbf{V}^{[j]})$  has a Jacobian matrix, a steepest decent direction and a step size associated with it. Since the algorithm is intuitive, and due to the space limitations, we have omitted most of the mathematical derivations (for more details we refer the reader to Part II, Section D of [15] and to [8]). The full algorithm is presented in Table 1. Below we comment on some of its steps.

Step 1: By comparing (9) & (11), it can be easily verified that

$$\mathbf{D}\Psi_R^{[j]} = \sum_{k=1, k \neq j}^K \mathbf{D}\mathbf{L}_s^{[kj]} \mathbf{D}\mathbf{Q}_R^{[kj]} \text{ and } \mathbf{D}\Psi_I^{[j]} = \sum_{k=1, k \neq j}^K \mathbf{D}\mathbf{L}_s^{[kj]} \mathbf{D}\mathbf{Q}_I^{[kj]} \quad (10)$$

Step 3 is required for steps 4 & 5. Note that the  $GS\{\cdot\}$  operator denotes the Gram-Schmidt Orthogonalization of a matrix, and is described in detail in [11].

Steps 4,5: They can be thought of as "calibration steps" for the algorithm step size (Armijo's rule), to ensure convergence (initially presented in [12], then later used by [8]). In a nutshell, step 4 ensures that the choice of  $\gamma$  will significantly reduce the function while step 5 ensures that the step is not too big to overshoot the optimal point.

Steps 6,7: Step 6 is simple and intuitive. However, the newly computed solution  $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}$  does not satisfy the unitary constraints. This can be effectively accomplished by using step 7 that will project the solution back onto the constraint surface (the surface of a hypersphere) [11].

Step 8: It was shown in [8] that

$$df = \begin{bmatrix} \mathbf{D}\mathbf{L}_s^{[1]} & \dots & \mathbf{D}\mathbf{L}_s^{[K]} \\ & & \vdots \\ \mathbf{D}\mathbf{Q}_R^{[K1]} & \dots & 0 \end{bmatrix} \begin{bmatrix} \text{vec}(d\mathbf{V}_R^{[1]}) \\ \vdots \\ \text{vec}(d\mathbf{V}_R^{[K]}) \end{bmatrix} + \begin{bmatrix} \mathbf{D}\mathbf{L}_s^{[1]} & \dots & \mathbf{D}\mathbf{L}_s^{[K]} \\ & & \vdots \\ \mathbf{D}\mathbf{Q}_I^{[K1]} & \dots & 0 \end{bmatrix} \begin{bmatrix} \text{vec}(d\mathbf{V}_I^{[1]}) \\ \vdots \\ \text{vec}(d\mathbf{V}_I^{[K]}) \end{bmatrix} \quad (11)$$

```

Start with random  $\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}$ 
for  $n = 1, 2, \dots$ 
  for  $j = 1, \dots, K$ 
    1. Compute the corresponding Jacobian matrix  $D\Psi_R^{[j]}$  and  $D\Psi_I^{[j]}$ 
    2. Find steepest descent direction  $\text{vec}(\mathbf{Z}^{[j]}) = - (D\Psi_R^{[j]} + iD\Psi_I^{[j]})'$ 
    3. Find  $\mathbf{P}_1^{[j]} = GS\{\mathbf{V}^{[j]} + 2\gamma^{[j]}\mathbf{Z}^{[j]}\}$ 
        $\mathbf{P}_2^{[j]} = GS\{\mathbf{V}^{[j]} + \gamma^{[j]}\mathbf{Z}^{[j]}\}$ 
    4. while  $(f(\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}) - f(\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[j-1]}, \mathbf{P}_1^{[j]}, \mathbf{V}^{[j+1]}, \dots, \mathbf{V}^{[K]})$ 
        $\geq 2\gamma^{[j]} < \mathbf{Z}^{[j]}, \mathbf{Z}^{[j]} >)$ 
       then  $\gamma^{[j]} = 2\gamma^{[j]}, \mathbf{P}_1^{[j]} = GS\{\mathbf{V}^{[j]} + 2\gamma^{[j]}\mathbf{Z}^{[j]}\}$ 
    5. while  $(f(\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[K]}) - f(\mathbf{V}^{[1]}, \dots, \mathbf{V}^{[j-1]}, \mathbf{P}_2^{[j]}, \mathbf{V}^{[j+1]}, \dots, \mathbf{V}^{[K]})$ 
        $< \gamma^{[j]} < \mathbf{Z}^{[j]}, \mathbf{Z}^{[j]} >)$ 
       then  $\gamma^{[j]} = \gamma^{[j]} / 2, \mathbf{P}_2^{[j]} = GS\{\mathbf{V}^{[j]} + \gamma^{[j]}\mathbf{Z}^{[j]}\}$ 
    6.  $(\mathbf{V}^{[j]})^{n+1} = (\mathbf{V}^{[j]})^n + \gamma^{[j]}\mathbf{Z}^{[j]}$ 
    7.  $(\mathbf{V}^{[j]})^{n+1} = GS\{(\mathbf{V}^{[j]})^{n+1}\}$ 
    8. if  $(| < \mathbf{Z}^{[j]}, \mathbf{Z}^{[j]} > | < \text{desired tolerance})$  then exit
  end
end
end

```

**Table 1: Proposed SD Algorithm**

$\langle \mathbf{Z}, \mathbf{Z} \rangle = \Re\{\text{tr}(\mathbf{Z}^H (\mathbf{I} - 1/2\mathbf{X}\mathbf{X}^H)\mathbf{Z})\}$  where  $\mathbf{X}$  is the optimization variable and  $\mathbf{Z}$  is the SD direction. Exploiting the fact that  $\text{tr}(\mathbf{A}^H \mathbf{X}) = \text{vec}(\mathbf{A})^H \text{vec}(\mathbf{X})$ , we show that:

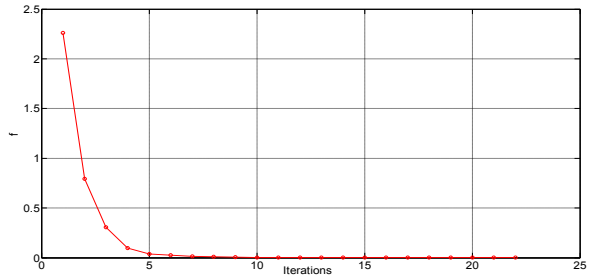
$$\langle \mathbf{Z}^{[j]}, \mathbf{Z}^{[j]} \rangle = \Re\{\text{vec}(\mathbf{Z}^{[j]})^H (\mathbf{I}_d \otimes (1/2)\mathbf{V}^{[j]}\mathbf{V}^{[j]H}) \text{vec}(\mathbf{Z}^{[j]})\} \quad (12)$$

where  $\otimes$  denotes the Kronecker product for matrices.

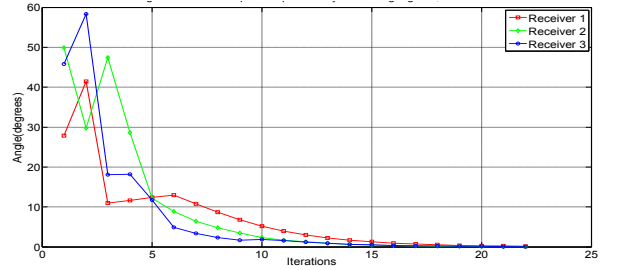
#### IV. NUMERICAL RESULTS

Before we move to presenting our simulations, we remind the reader that we do not expect our approach to outperform other ones (Section I). Rather, we want to ensure that this one-sided approach does not entail any significant loss in performance compared to [2].

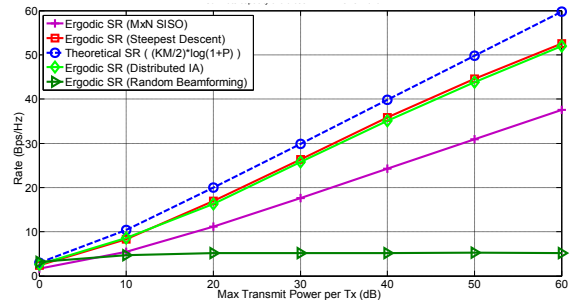
First, we simulated a 3-user  $2 \times 2$  MIMO IC where the desired DoFs per user are 1 ( $d = 1$ ), and AWGN noise with unit variance is assumed at each receiver. As we can see, in this given example (Fig. 3), our cost function drops steadily; a clear indication of the validity of our approach. We note that more sophisticated algorithms can be used for faster and more accurate convergence, such as Newton-type methods [8] or RLS-type algorithms [16]. However, we still have to check if the final solution satisfies the IA conditions. Since we have two interfering signals at each receiver, we can plot the angle between the spaces spanned by each interfering signal, at each receiver. Evidently, a perfect IA solution is characterized by a zero angle at each receiver. Referring to Fig. 4, the algorithm decreases the subspace between the two interfering signals at each receiver to zero (asymptotically to the iteration count). We also compare the performance of this algorithm to the Distributed IA algorithm in [2]. In Fig.5, the dashed line is the theoretical sum-rate capacity predicted by IA, i.e.



**Figure 3: Cost function**



**Figure 4: Angle between spaces spanned by interfering signals, at each Rx**



**Figure 5: Ergodic Sum-Rate (SR) capacity of a 3-user  $2 \times 2$  MIMO IC with  $d = 1$**

$C_{SR} = (KM/2)\log(1+P)$ . As we can see, the curves that correspond to our algorithm and Distributed IA (with square and diamond shaped points, respectively) are almost overlapping indicating the same performance. Moreover, we notice that at high SNR, they follow the linear scaling predicted by IA, i.e. they have same slope as the theoretical curve (almost – due to simulation resolution limitations), thus verifying that they indeed succeed in aligning the interference.

#### Remark:

To satisfy the IA conditions, each receiver must be able to partition its received  $N$ -dimensional signal space into two subspaces of equal size (i.e.,  $d = N/2$ ) where each user is able to achieve  $M/2$  DoFs (for a total of  $KM/2$ , the SR capacity of the  $K$ -user MIMO IC). However, for cases where the alignment is not feasible, i.e.  $K > 3$ , the algorithm still creates an interference-free  $d$ -dimensional subspace but with  $d < N/2$ . Thus, even in this suboptimal case, each user is still able to achieve  $d$  DoFs (where  $0 < d < N/2$ , for a total that is strictly less than  $KM/2$ ), a clear improvement over the interference limited case ( $d = 0$ ).

#### V. DISCUSSION

In this section we establish equivalences between our approach and [2]. Interestingly, it is straightforward and

intuitive to mathematically verify that our cost function is nothing but the Weighted Leakage Interference (WLI) in [2] (the proof is presented in the Appendix of [15]). As a matter of fact, our algorithm moves along a lower dimensional cut of the WLI function (see footnote) that contains the local / global optimum. Consequently, they both converge to the same local / global minimum (when initialized with the same point). Thus, the following points are common grounds for comparison:

- In terms of convergence, our algorithm aligns interference in a matter similar to Distributed IA, i.e. it converges to an IA solution when it is feasible (the fact that the cost function decreases with every iteration, is an inherent property of SD algorithms). However, as is the case in [2], convergence to a global optimum is not guaranteed, because even though our cost function may be convex under certain assumptions, the unitary constraints by themselves form a non-convex set. The *global* convergence of IA algorithms is still an open problem.
- Simulations show that there is no loss in capacity. This result is further backed by the fact that our approach and [2] are both subspace methods, i.e. they both ignore noise when finding the subspace with the least interference (optimal at high INR only).

However, in a realistic setting, the following are seen as clear benefits of our approach:

- We removed  $U^{[1]}, \dots, U^{[K]}$  from the minimization. Thus, the algorithm can run at the transmitters only, getting around the need to alternate between forward and reverse network (an inherent property of alternating minimization algorithms) which can induce a significant overhead in mobile environments.
- The algorithm is completely transparent to the receivers, i.e. they are oblivious to the entire process. All they have to do is “tune” to the subspace that has the lowest interference (not more than what today’s receivers are doing). The purpose of this entire approach is to ensure that this subspace has minimal interference.

We note that a typical drawback of such IA techniques is the need for global channel knowledge at the transmitters. However, this assumption is not unrealistic since recent results such as [13] and [14] show that under certain mild assumptions, limited feedback can achieve the full sum-rate degrees-of-freedom of the  $K$ -user MIMO/SISO IC.

## VI. CONCLUSION AND FUTURE WORK

To sum up, we proposed in this paper an algorithm for IA that approaches the problem from the transmitter’s side. In that sense, it improves many of the shortcomings of Distributed IA in [2] by getting around the need to alternate between the forward and reverse network (thereby avoiding unnecessary overhead), by being transparent to the receivers, and by alleviating the channel reciprocity assumption (making it equally applicable to TDD and FDD systems). This was achieved using the mathematical formulations that we derived to account for interference in the network, and then using

matrix differentials to mathematically express the differential of our cost function, resulting in a Jacobian-like expression. Moreover, we used the latter to apply a simple SD algorithm to the cost function, and showed that it converges to an IA solution (when feasible). Furthermore, mathematical analysis proved that the two approaches are equivalent in terms of cost functions and convergence. Finally, our simulations showed that they exhibit the same performance. In the future, we wish to further investigate the issue of global convergence, as well as increased convergence speed for one-sided IA algorithms.

## ACKNOWLEDGMENT

The authors wish to acknowledge the many helpful discussions with Professor Spyridon Vassilaras, of AIT’s B-WiSE Lab.

## REFERENCES

- [1] V. R. Cadambe and S. A. Jafar, “Interference alignment and degrees of freedom of the  $K$ -user interference channel” *IEEE Trans. Information Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008
- [2] K. Gomadam, V. R. Cadambe, and S. A. Jafar, “Approaching the capacity of wireless networks through distributed interference alignment”, available on arXiv.
- [3] Cadambe, Jafar, “Reflections on interference alignment and the degrees of freedom of the  $K$ -user MIMO interference channel”, *IEEE Inform Theory Society Newsletter*, Vol. 54, No. 4, Dec. 09, pp. 5 - 8.
- [4] M. Maddah-Ali, A. Motahari, and A. Khandani, “Signaling over MIMO multi-base systems—Combination of multi-access and broadcast schemes” in *Proc. Int. Symp. Inform Theory 2006*, pp. 2104–2108.
- [5] M. A. Maddah-Ali, A. S. Motahari, and A. K. Khandani, “Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis” *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3457–3470, Aug. 2008.
- [6] S. W. Peters, R. W. Heath, “Interference alignment via alternating minimization”, *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, April 2009, pp. 2445 – 2448.
- [7] S. W. Peters, R. W. Heath, Jr, “Cooperative algorithms for MIMO interference channels”, to appear in *IEEE Trans. On Veh. Tech.* Preprint available on arXiv.
- [8] J. H. Manton, “Optimization algorithms exploiting unitary constraints”, *IEEE Trans. Signal Process.* 50 (2002), no. 3, 635–650.
- [9] J. Mangus, “On differentiating eigenvalues and eigenvectors”, *Econometric Theory*, Vol. 1, No. 2 (Aug., 1985), pp. 179-191.
- [10] Magnus, J. R. and H. Neudecker “Matrix differential calculus with applications in statistics and econometrics”, *John Wiley & Sons*: Chichester/New York, reprinted 1990, pp. 198-200
- [11] C. B. Papadias, “Globally convergent blind source separation based on a multiuser kurtosis maximization Criterion” *IEEE Trans. Signal Processing*, vol. 48, No. 12, pp. 3508-3519, Dec. 2000
- [12] E. Polak, “Optimization: Algorithms and consistent approximations”. New York: Springer-Verlag, 1997
- [13] R. T. Krishnamachari, M. K. Varanasi “Interference alignment under limited feedback for MIMO ICs”, *Proc. of ISIT 2009, Seoul, Korea*
- [14] J. Thurkal, H. Bolekskei, “Interference Alignment under limited feedback”, *Proc. ISIT 2009, Seoul, Korea*, pp. 1759-1763
- [15] H. Ghauch, “Interference alignment for multiuser cellular networks with multi-antenna terminals”, *Master’s Thesis, Carnegie Mellon University*, April 2011, available upon request in CMU Library, and online [www.contrib.andrew.cmu.edu/~hgg/docs/Thesis%20Report.pdf](http://www.contrib.andrew.cmu.edu/~hgg/docs/Thesis%20Report.pdf)
- [16] C. B. Papadias, “Globally convergent algorithms for blind source separation”, *X Europ. Sign. Proc. Conf. (EUSIPCO 2000)*, Tampere Finland, Sept 4-8 2000.