HOMOTOPY CONTINUATION FOR VECTOR SPACE INTERFERENCE ALIGNMENT IN MIMO X NETWORKS

Óscar González, Jacobo Fanjul[†] and Ignacio Santamaría**

Communications Engineering Dept., University of Cantabria, Santander, Spain
e-mail: *{oscargf,nacho}@gtas.dicom.unican.es, [†]jacobo.fanjul@alumnos.unican.es

ABSTRACT

In this paper we propose an algorithm to design interference alignment (IA) precoding and decoding matrices for MIMO X networks (XN). The proposed algorithm is rooted in the homotopy continuation techniques commonly used to solve systems of nonlinear equations. Homotopy methods find the solution of a target system by smoothly deforming the known solutions of a start system which can be trivially solved. The key observation leading to a simple start system is realizing that the *inverse IA problem*, i.e., finding the channels that satisfy the IA conditions given a set of precoders and decoders, is linear and, therefore, a convenient trivial system. Once the start system has been solved, standard prediction and correction techniques are applied to track the solution all the way to the target system. Our results show that the proposed algorithm is able to consistently find solutions achieving the maximum number of degrees of freedom (DoF) whereas alternating minimization techniques, which typically work well for the interference channel (IC), repeatedly fail for the XN. Further, the algorithm provides insights into the feasibility of alignment in MIMO X networks for which theoretical results are scarce.

Index Terms— Degrees of freedom, homotopy continuation, interference alignment, MIMO, X network.

1. INTRODUCTION

The concept of interference alignment (IA) originated out of the study of the degrees of freedom (DoF) of the 2-user X channel [1, 2] and shortly afterwards it was extended to the K-user interference channel (IC) [3] that caught the greatest deal of attention owing to its simpler formulation when compared to an X network. An $M \times N$ multiple-input multiple-output (MIMO) X network represents the most general single-hop network with M transmitters and N receivers, each of them equipped with multiple antennas. Understanding X networks (XNs) is of paramount importance since this general scenario subsumes many other well-known network topologies such as the multiple-access, broadcast, interference, X, and interference broadcast channel.

In spite of its generality, existing results for X networks are scarce, most of them focusing on the 2-user X channel. In particular, the total number of degrees of freedom (DoF) when both users are equipped with the same number of antennas, along with an outer bound for the asymmetric 2-user X channel, were obtained in [1]. Recently, the authors in [4] proposed a scheme achieving the aforementioned bound. A DoF bound for the $M \times N$-user XN where

all users are equipped with A antennas and have symmetric message demands was proposed and shown to be achievable in [5] and [6], respectively. In the particular case of asymmetric message demands, the results in [7] apply. Finally, [6] extended the *properness* condition in [8] to provide an upper bound on the linear DoF without channel extensions, although the tightness of the proposed bound was not evaluated.

The list of algorithms to compute IA solutions for X networks is also very scarce. First, Jafar and Shamai [1] proposed the so-called JS-scheme for the 2-user MIMOX channel which is able to achieve the outer bound for some particular scenarios. Later, Agustín and Vidal [4, 9] presented an algorithm based on the generalized singular value decomposition that attains the outer bound on the degrees of freedom (DoF) region for any antenna configuration and number of channel extensions. This algorithm is again limited to the 2-user MIMO X channel. Some other algorithms, which were originally developed for the MIMO IC, can be straightforwardly ported to the XN. That is the case of the well-known alternating minimization algorithm in [10] which always finds a solution in the IC when the system is feasible, but repeatedly fails in the XN. The reason for this is that in the XN every link acts as both a desired and an interfering link and, due to this coupling, the alternating minimization algorithm is not able to guarantee the rank of the signal in the desired links at the same time it minimizes the interference leakage.

In this paper we propose an algorithm to compute interference alignment solutions for the asymmetric X network with no channel extensions. It is based on homotopy continuation, a numerical method which is widely used to solve multivariate systems of nonlinear equations. The proposed algorithm extends the work for single-beam ICs in [11] to multi-beam XNs. The main novelty is how the starting point for the homotopy continuation procedure is computed which happens to be as simple as solving a system of linear equations. Our algorithm focuses on finding the maximum DoF solution with the highest success rate and no effort of maximizing the sum-rate or optimizing any other figure of merit is made. We show, that the algorithm is able to find the maximal DoF solution with high probability and clearly outperforms the algorithm in [10] in terms of achieved DoF.

Notation: Uppercase (lowercase) boldface letters will be used for matrices (column vectors); $(\cdot)^H$ for conjugate transpose (Hermitian) and $(\cdot)^+$ for the matrix Moore-Penrose pseudoinverse. $\otimes$ denotes the Kronecker product and $\mathbf{I}_{m,n}$ and $\mathbf{0}_{m,n}$ the $m \times n$ identity and all-zero matrices, respectively. The Jacobian matrix of a matrix function $\mathbf{G}(\mathbf{A})$ with respect to a matrix $\mathbf{A}$ will be denoted as $D\mathbf{G}(\mathbf{A})$ following the definition in [12] and we will write an increment in variable $\mathbf{A}$ as $\Delta\mathbf{A}$. Further, we define the operator

$$\operatorname*{cat}_{s \in \mathcal{S}}(\mathbf{A}_s)$$

as the horizontal concatenation of the indexed matrices $\mathbf{A}_s$ where the members s of the set S are taken in reverse lexicographic order.

2. SYSTEM MODEL

Let us focus on the symmetric K-user MIMO X network[1] where all transmitters and receivers are equipped with A and B antennas, respectively, and each transmitter ℓ wishes to send d independent data streams to each receiver k. Let now $\mathbf{V}_{j\ell} \in \mathbb{C}^{A \times d}$ be the precoding matrix that transmitter ℓ will use to send its information to receiver j where $j, \ell \in \{1, \ldots, K\}$. At the other end of each link, receiver $k \in \{1, \ldots, K\}$ applies a decoding matrix $\mathbf{U}_k \in \mathbb{C}^{B \times Kd}$. Then, the signal after decoding at the k-th receiver can be modeled as

$$\mathbf{r}_k = \mathbf{U}_k^H \left(\sum_\ell \mathbf{H}_{k\ell}\mathbf{V}_{k\ell}\mathbf{s}_{k\ell} + \sum_\ell \sum_{j \neq k} \mathbf{H}_{k\ell}\mathbf{V}_{j\ell}\mathbf{s}_{j\ell} + \mathbf{n}_k \right) \quad (1)$$

where $\mathbf{s}_{k\ell} \in \mathbb{C}^d$ contains the information symbols that transmitter ℓ wishes to send to receiver k, $\mathbf{H}_{k\ell} \in \mathbb{C}^{B \times A}$ is the flat-fading MIMO channel from transmitter ℓ to receiver k and $\mathbf{n}_k \in \mathbb{C}^B$ is the additive and spatially white Gaussian noise at receiver k.

In order to avoid undesired interference, transmitters must design their precoders to confine the interference to a reduced-dimensionality subspace in such a way it can be zero-forced by receivers with their corresponding decoding matrices. This transmission strategy is known as interference alignment (IA) and the existence of such a precoder design requires the simultaneous satisfiability of the following conditions:

$$\mathbf{U}_k^H \mathbf{H}_{k\ell}\mathbf{V}_{\bar{k}\ell} = \mathbf{0}, \quad \forall k, \ell \qquad (2)$$

$$\text{rank}\left(\mathbf{U}_k^H \left[\mathbf{H}_{k1}\mathbf{V}_{k1}, \ldots, \mathbf{H}_{kK}\mathbf{V}_{kK} \right] \right) = Kd, \quad \forall k \qquad (3)$$

where $\mathbf{V}_{\bar{k}\ell}$ is defined as the horizontal concatenation of all $\mathbf{V}_{j\ell}$ such that $j \neq k$, i.e.,

$$\mathbf{V}_{\bar{k}\ell} \overset{\text{def}}{=} \underset{j \neq k}{\text{cat}}\left(\mathbf{V}_{j\ell} \right).$$

Equation (2) guarantees that all interferences are properly zero-forced and (3) preserves the desired signal dimensionality at all receivers. We note that these conditions are analogous to those that were put forward in [8, 10] in the context of the interference channel. However, there exists an important difference which makes this a much harder to solve problem. More specifically, in an IC, transmitters only share a message with their corresponding receivers. Then, (2) and (3) simplify into

$$\mathbf{U}_k^H \mathbf{H}_{k\ell}\mathbf{V}_{\ell\ell} = \mathbf{0}, \quad \forall k, \forall \ell \neq k \qquad (4)$$

$$\text{rank}\left(\mathbf{U}_k^H \mathbf{H}_{kk}\mathbf{V}_{kk} \right) = d, \quad \forall k \qquad (5)$$

where (5) is, in addition, automatically satisfied as long as both precoders and decoders are guaranteed to be full column rank. This is clear by taking into consideration that the channels appearing in (5) are independent of those appearing in (2). The alternating minimization algorithm in [10] exploits this fact and solves (4) by restricting the precoders and decoders to lie in the Stiefel manifold, that is, $\mathbf{V}_{\ell\ell}^H \mathbf{V}_{\ell\ell} = \mathbf{I}$ and $\mathbf{U}_k^H \mathbf{U}_k = \mathbf{I}$. Unfortunately, this is not the case

[1]For the sake of conciseness we restrict our explanation to an X network with the same number of transmitters and receivers, i.e. $M = N = K$, involving only unicast traffic, i.e. each transmitted message is only demanded by a single receiver. However, the presented results apply in full generality.

for the MIMO X network since the same channel matrices appear in both (2) and (3).

In order to provide some insight into the necessary conditions for (2) to be solvable we follow the common arguments in the literature [6, 8] analyzing precoder and decoder invariances. It is clear that given the sets of precoders and decoders, $\{\mathbf{V}_{j\ell}\}$ and $\{\mathbf{U}_k\}$, satisfying (2), we can right-multiply them by arbitrary invertible matrices and (2) still holds. Therefore, a total of d^2 (K^2d^2) elements can be arbitrarily fixed in each precoder (decoder) leaving a total of $N_v = K^2d[A + B - (K + 1)d]$ free variables. The total number of scalar equations in (2) relating those variables are $N_e = K^3(K-1)d^2$. Therefore, a necessary condition [6] for (2) to be solvable is $N_e \leq N_v$ or, equivalently,

$$d \leq \frac{A+B}{K^2+1}. \qquad (6)$$

A system satisfying (6) is said to be *proper* [8] which does not necessarily imply it is feasible although, usually, it is a good indicator of feasibility. On the contrary, an *improper* system is always infeasible [6, 8, 13, 14].

3. HOMOTOPY CONTINUATION FOR VECTOR SPACE IA

Homotopy continuation is a numerical method for solving systems of nonlinear equations which is based on the idea of defining a parametrized transformation or homotopy that gradually deforms a trivially solvable system or *start* system into the target system we want to solve [15, 16]. One simple and widely used transformation is a convex combination of both systems, the so-called *convex homotopy*. In the particular case of IA, it leads to consider a system as the one in (2) where the MIMO channels are obtained as a convex combination of a *start* channel, $\overline{\mathbf{H}}_{k\ell}$, and the target channel, $\mathbf{H}_{k\ell}$. The combination is controlled by the continuation parameter, t, which leads to a homotopy function $\mathbf{G}_{k\ell}(\mathbf{U}_k^H, \mathbf{V}_{\bar{k}\ell}, t)$ defined as

$$\mathbf{U}_k^H \underbrace{\left((1-t)\overline{\mathbf{H}}_{k\ell} + t\mathbf{H}_{k\ell} \right)}_{\mathbf{H}_{k\ell}(t)} \mathbf{V}_{\bar{k}\ell}, \quad \forall k, \ell \text{ and } t \in [0, 1]. \qquad (7)$$

Our goal is to move along the path $\mathbf{G}_{k\ell}(\mathbf{U}_k^H, \mathbf{V}_{\bar{k}\ell}, t) = \mathbf{0} \; \forall k, \ell$, from $t = 0$ to $t = 1$ in small steps, Δt. Numerically tracing this path is usually accomplished by a predictor/corrector method [17]. This method consists of two basic steps which are executed iteratively, namely, the prediction and correction steps. In particular, a first order approximation of the homotopy function

$$\mathbf{G}_{k\ell}(\mathbf{U}_k^H + \Delta\mathbf{U}_k^H, \mathbf{V}_{\bar{k}\ell} + \Delta\mathbf{V}_{\bar{k}\ell}, t + \Delta t) = \mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\mathbf{V}_{\bar{k}\ell} + \\ \Delta\mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\mathbf{V}_{\bar{k}\ell} + \mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\Delta\mathbf{V}_{\bar{k}\ell} + \\ \mathbf{U}_k^H (\mathbf{H}_{k\ell} - \overline{\mathbf{H}}_{k\ell})\mathbf{V}_{\bar{k}\ell}\Delta t \quad \forall k, \ell, \qquad (8)$$

gives rise to the basic Euler prediction and Newton correction. Assuming we have a point $(\{\mathbf{U}_k\}, \{\mathbf{V}_{j\ell}\}, t)$ near the path (i.e. $\mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\mathbf{V}_{\bar{k}\ell} \approx \mathbf{0} \; \forall k, \ell$), we may predict to an approximate solution at $t + \Delta t$ by setting $\mathbf{G}_{k\ell}(\mathbf{U}_k^H + \Delta\mathbf{U}_k^H, \mathbf{V}_{\bar{k}\ell} + \Delta\mathbf{V}_{\bar{k}\ell}, t + \Delta t) = \mathbf{0}$:

$$\Delta\mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\mathbf{V}_{\bar{k}\ell} + \mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\Delta\mathbf{V}_{\bar{k}\ell} = \\ -\mathbf{U}_k^H (\mathbf{H}_{k\ell} - \overline{\mathbf{H}}_{k\ell})\mathbf{V}_{\bar{k}\ell}\Delta t \quad \forall k, \ell. \qquad (9)$$

Increments $\Delta\mathbf{V}_{\bar{k}\ell}$ and $\Delta\mathbf{U}_k^H \; \forall k, \ell$ are obtained by solving the system of linear equations in (9). Specific details on this are relegated to Section 3.1.

On the other hand, if the current point $(\{\mathbf{U}_k\}, \{\mathbf{V}_{j\ell}\}, t)$ is not as close to the path as we would like, i.e. the entries of $\mathbf{G}_{k\ell}(\mathbf{U}_k^H, \mathbf{V}_{\bar{k}\ell}, t)$ are larger than a predefined tolerance, we can hold t constant by setting $\Delta t = 0$ and obtain the Newton correction step:

$$\Delta\mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\mathbf{V}_{\bar{k}\ell} + \mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\Delta\mathbf{V}_{\bar{k}\ell} = -\mathbf{U}_k^H \mathbf{H}_{k\ell}(t)\mathbf{V}_{\bar{k}\ell}, \quad \forall k, \ell \tag{10}$$

Analogously to the prediction step, precoder and decoder updates, $\Delta\mathbf{V}_{\bar{k}\ell}$ and $\Delta\mathbf{U}_k^H \; \forall k, \ell$, are obtained by solving the system of linear equations in (10) (cf. Section 3.1) which leads to a new set of precoders and decoders, $\{\mathbf{V}_{j\ell} + \Delta\mathbf{V}_{j\ell}\}$ and $\{\mathbf{U}_k + \Delta\mathbf{U}_k\}$, which are closer to the tracked path.

3.1. Some implementation details

In this section we provide supplementary material that may be helpful for the practical implementation of (9) and (10). First, it is convenient to define a vector $\mathbf{w} = \left[\underset{(j,\ell)}{\mathrm{cat}}((\mathrm{vec}\,\mathbf{V}_{j\ell})^T), \underset{k}{\mathrm{cat}}((\mathrm{vec}\,\mathbf{U}_k^H)^T)\right]^T$ by simply stacking up all precoder and decoder elements. Both (9) and (10) describe systems of coupled linear equations which can be conveniently solved if regarded as large sparse linear systems. We follow up with the analysis of (9). Our goal is to write the set of linear equations in (9) as a single linear equation, $D\mathbf{G}(\mathbf{w})\Delta\mathbf{w} = -D\mathbf{G}(t)$. In order to do so, we first vectorize (9) leading to

$$\underbrace{\left(\underset{j \neq k}{\mathrm{cat}}(\mathbf{V}_{j\ell}^H \mathbf{H}_{k\ell}^H) \otimes \mathbf{I}_{Kd}\right)}_{D\mathbf{G}_{k\ell}(\mathbf{U}_k^H)}\Delta\mathrm{vec}\,\mathbf{U}_k^H + \sum_{j \neq k}\underbrace{\left(\mathbf{P}_j \otimes (\mathbf{U}_k^H \mathbf{H}_{k\ell})\right)}_{D\mathbf{G}_{k\ell}(\mathbf{V}_{j\ell})}\Delta\mathrm{vec}\,\mathbf{V}_{j\ell}$$

$$= -\underbrace{\mathrm{vec}\left(\mathbf{U}_k^H(\mathbf{H}_{k\ell} - \overline{\mathbf{H}}_{k\ell})\mathbf{V}_{\bar{k}\ell}\right)}_{D\mathbf{G}_{k\ell}(t)}\Delta t, \quad \forall k, \ell, \tag{11}$$

where $\mathbf{P}_j$ is the j-th block of d columns in $\mathbf{I}_{Kd}$.[2] Once the equations have been vectorized, we can stack them together to show that the Euler prediction step amounts to solve the following sparse linear system in a least-squares sense:

$$D\mathbf{G}(\mathbf{w})\Delta\mathbf{w} = -D\mathbf{G}(t)\Delta t \;\Rightarrow\; \Delta\mathbf{w} = -D\mathbf{G}(\mathbf{w})^+ D\mathbf{G}(t)\Delta t, \tag{12}$$

where $D\mathbf{G}(\mathbf{w})$ is the Jacobian matrix of the system of matrix equations (2), which comprises all the derivatives with respect to the variables in $\{\mathbf{V}_{j\ell}\}$ and $\{\mathbf{U}_k^H\}$ in the order specified in (15). It is a block partitioned matrix with as many row partitions as channel matrices and as many column partitions as precoding and decoding matrices. Sparsity comes from the fact that each equation involves a subset of the variables and, therefore, many blocks in (15) are zero. Specifically, $D\mathbf{G}_{k\ell}(\mathbf{V}_{jp}) = \mathbf{0}$ when $j = k$ or $p \neq \ell$, and $D\mathbf{G}_{k\ell}(\mathbf{U}_k^H) = \mathbf{0}$ when $p \neq k$. The solution vector $\Delta\mathbf{w}$ contains the update values for all the variables in both precoders and decoders and the derivate with respect to the continuation parameter is built from all partial derivatives as $D\mathbf{G}(t) = \underset{(k,\ell)}{\mathrm{cat}}(D\mathbf{G}_{k\ell}(t)^T)^T$.

Similarly, the Newton correction step can be written as the solution to a single linear equation, $D\mathbf{G}(\mathbf{w})\Delta\mathbf{w} = -\mathbf{g}$, which can be obtained by vectorizing and stacking up all the equations in (10):

$$D\mathbf{G}_{k\ell}(\mathbf{U}_k^H)\Delta\mathrm{vec}\,\mathbf{U}_k^H + D\mathbf{G}_{k\ell}(\mathbf{V}_{j\ell})\Delta\mathrm{vec}\,\mathbf{V}_{j\ell}$$
$$= -\underbrace{\mathrm{vec}\left(\mathbf{U}_k^H \mathbf{H}_{k\ell}\mathbf{V}_{\bar{k}\ell}\right)}_{\mathbf{g}_{k\ell}}\Delta t, \quad \forall k, \ell, \tag{13}$$

---

[2] We omit the details due to space constraints but basic linear algebra and the identity $\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\,\mathrm{vec}(\mathbf{B})$ suffice to get (11) and (13).

Algorithm 1: Interference alignment via homotopy continuation in MIMO X networks.

Input: $\{\mathbf{H}_{k\ell}\}$, Δt, `NwtTol`, `MaxNwtIter`, `MinStepSize`, `NumHitsToDoubleStep`
Output: Sets $\{\mathbf{V}_{j\ell}\}$ and $\{\mathbf{U}_k\}$ satisfying (2) and a convergence indicator, PathFailed
/* Inverse IA   */
Obtain $\{\overline{\mathbf{H}}_{k\ell}\}$ as shown in (16)
$t = 0$, NumHits = 0, PathFailed = `false`
$\mathbf{w} = \left[\underset{(j,\ell)}{\mathrm{cat}}((\mathrm{vec}\,\mathbf{V}_{j\ell})^T), \underset{k}{\mathrm{cat}}((\mathrm{vec}\,\mathbf{U}_k^H)^T)\right]^T$
$t^* = t$, $\mathbf{w}^* = \mathbf{w}$ // backup variables
while $t < 1$ **do**
  $t = \min(t + \Delta t, 1)$
  /* Euler prediction   */
  $\mathbf{w} = \mathbf{w} - D\mathbf{G}(\mathbf{w})^+ D\mathbf{G}(t)\Delta t$ as indicated in (12)
  /* Newton correction   */
  NewtonFailed = `true`
  for *iter* = 1 **to** *MaxNwtIter* **do**
    $\mathbf{w} = \mathbf{w} - D\mathbf{G}(\mathbf{w})^+\mathbf{g}$ as shown in (14)
    if $\|\mathbf{g}\|^2 < $ *NwtTol* **then**
      NumHits = NumHits + 1
      NewtonFailed = `false`
      break
  /* Step size adaptation routine   */
  if *NumHits* == *NumHitsToDoubleStep* **then**
    $\Delta t = 2\Delta t$
    $t^* = t$, $\mathbf{w}^* = \mathbf{w}$, NumHits = 0
  else if *NewtonFailed* **then**
    $\Delta t = \Delta t/2$
    $t = t^*$, $\mathbf{w} = \mathbf{w}^*$, NumHits = 0
    if $\Delta t < $ *MinStepSize* **then**
      PathFailed = `true`
      return
return

---

and, therefore, stacking up all the equations

$$D\mathbf{G}(\mathbf{w})\Delta\mathbf{w} = -\mathbf{g} \;\Rightarrow\; \Delta\mathbf{w} = -D\mathbf{G}(\mathbf{w})^+\mathbf{g}, \tag{14}$$

where $\mathbf{g} = \underset{(k,\ell)}{\mathrm{cat}}(\mathbf{g}_{k\ell})$.

Given that Newton method converges quadratically to a point in the path, a common strategy is to run the correction step several times establishing a limit on the number of executions to a maximum of `MaxNwtIter` or until all the entries of $\mathbf{g}$ are below the predefined tolerance `NwtTol`, whatever happens first. As a final remark, in order to guarantee the success of the path tracking procedure it is important to implement a step size adaptation rule. A simple rule will help to detect convergence of the Newton method to local minima and accelerate the execution of the path tracking routine. This is now explained in more detail.

It may happen that after a prediction step with a step size Δt we end up in the basin of attraction of a local minimum. In that case, the Newton method will exhaust the maximum number of allowed iterations, `MaxNwtIter`, without converging to the desired tolerance `NwtTol`. When this happens we consider that the Euler prediction has failed and repeat it with a smaller step size. A common criterion is to cut the step size in half. If in spite of implementing this step reduction procedure, a number of repeated failed predictions is

$$DG(\mathbf{w}) = \begin{bmatrix} D\mathbf{G}_{11}(\mathbf{V}_{11}) & D\mathbf{G}_{11}(\mathbf{V}_{21}) & \cdots & D\mathbf{G}_{11}(\mathbf{V}_{KK}) & D\mathbf{G}_{11}(\mathbf{U}_1^H) & \cdots & D\mathbf{G}_{11}(\mathbf{U}_K^H) \\ D\mathbf{G}_{21}(\mathbf{V}_{11}) & & & & & & \\ \vdots & & D\mathbf{G}_{k\ell}(\mathbf{V}_{jp}) & \vdots & \vdots & D\mathbf{G}_{k\ell}(\mathbf{U}_k^H) & \vdots \\ D\mathbf{G}_{KK}(\mathbf{V}_{11}) & \cdots & & D\mathbf{G}_{KK}(\mathbf{V}_{KK}) & D\mathbf{G}_{KK}(\mathbf{U}_1^H) & \cdots & D\mathbf{G}_{KK}(\mathbf{U}_K^H) \end{bmatrix} \tag{15}$$

obtained, it means that the path tracking procedure stagnated in a local minimum. This can be detected when the step size becomes too small. In particular, if it becomes smaller than a predefined minimum step size, `MinStepSize`, the whole path tracking procedure should be accounted as a failure. Conversely, if the correction step is successful for `NumHitsToDoubleStep` consecutive iterations, we can double the step size aiming to reduce the number of iterations of the path tracking routine. Additional implementation details can be found in Algorithm 1.

3.2. The inverse IA problem

We have described the routine which allows us to track the evolution of the solution matrices until the target system is reached. In this section we detail how to obtain the starting point of this procedure, the so-called start system. The key observation here is to consider what we call the *inverse IA* problem which is simply looking at (2) as if we were given $\mathbf{U}_k^H$ and $\mathbf{V}_{\bar{k}\ell}$ and we had to solve it for $\mathbf{H}_{k\ell}$. When regarded this way, the problem turns into a linear problem [18, Theorem 6.11] where all solutions can be parametrized as

$$\overline{\mathbf{H}}_{k\ell} = \mathbf{X}_{k\ell} - \mathbf{F}_k\mathbf{F}_k^H\mathbf{X}_{k\ell}\mathbf{G}_{\bar{k}\ell}\mathbf{G}_{\bar{k}\ell}^H \tag{16}$$

where $\mathbf{F}_k$ and $\mathbf{G}_{\bar{k}\ell}$ are orthonormal bases of $\mathbf{U}_k$ and $\mathbf{V}_{\bar{k}\ell}$, respectively, and $\mathbf{X}_{k\ell}$ is a non-zero arbitrary matrix. Therefore, in order to find a valid start system we just need to take $\mathbf{U}_k^H$, $\mathbf{V}_{\bar{k}\ell}$ and $\mathbf{X}_{k\ell}$ at random and then obtain $\overline{\mathbf{H}}_{k\ell}$ according to (16).

4. RESULTS

In this section, we compare the sum-rate performance of the proposed method (denoted as `HC`) against a generalization of the minimum interference leakage (`MinIL`) alternating minimization algorithm presented in [10], which has been conveniently adapted to work in MIMO X networks. For this comparison we have chosen four scenarios with the number of antennas A and B given in the second row of Table 1. Scenario 1 is the tightest scenario since it has the minimum number of antennas that are needed for the system to be feasible according to (6). The loosest scenario that we have considered is Scenario 4. The number of users for all scenarios is $K = 4$ and we ask both algorithms to compute a solution achieving $K^2d = 16$ DoF, that is, $d = 1$ streams per link. The results of 1000 independent channel realizations with unitary gain complex Gaussian entries were averaged.

As shown in Figure 1, `MinIL` shows a poor performance when the system is relatively tight. Its performance improves as the number of antennas is increased, requiring at least 11 antennas at both sides of the link to achieve the requested DoF. On the other hand, the `HC` algorithm provides DoF values which clearly overcome those achieved by `MinIL`. More specifically, the sum-rate in Scenario 1 is similar to that achieved by `MinIL` in Scenario 3, with the advantage of `HC` requiring fewer antennas. In fact, as shown in Table 1, `HC` is not only achieving a higher DoF, but also its probability of finding the maximum of 16 DoF is considerably higher.

Scenario	1	2	3	4
$(A, B) =$	$(8, 9)$	$(9, 9)$	$(10, 10)$	$(11, 11)$
HC	0.18	0.15	0.02	0.00
MinIL	1.00	1.00	0.56	0.01

Table 1. $\mathrm{Prob}[\mathrm{DoF} < K^2d = 16]$ or probability of not achieving the requested number of degrees of freedom for $K = 4$ and $d = 1$.
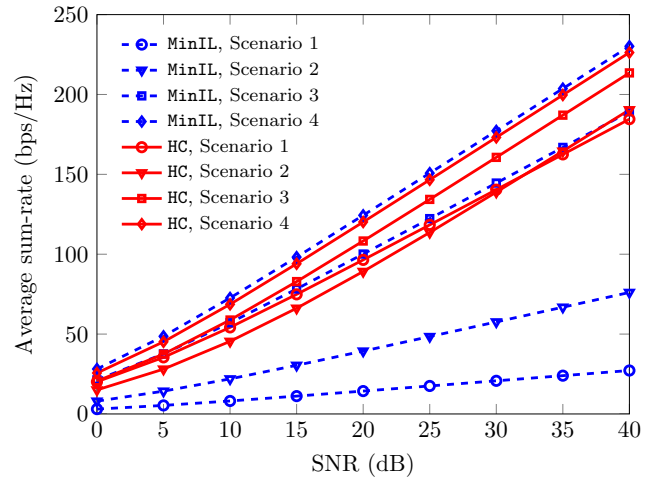

Fig. 1. Average sum-rate achieved by `MinIL` and `HC` in the four considered scenarios.

Finally, we note that the probability of attaining the maximum DoF in n executions of the algorithm grows as $1 - (\mathrm{Prob}[\mathrm{DoF} < K^2d])^n$. Therefore, given the low probability of failure of `HC`, the algorithm may be useful as a numerical means to obtain evidence of feasibility for a given scenario. On the other hand, the high failure probability of `MinIL` in some scenarios, e.g. Scenarios 1 and 2, would not allow us to make such a claim.

5. CONCLUSION

We have introduced an algorithm to compute interference alignment solutions in X networks with no channel extensions. It is based on homotopy continuation, a numerical method which is widely used to solve multivariate polynomial systems of nonlinear equations. We show that the proposed algorithm is able to consistently find solutions achieving the maximum number of degrees of freedom. When compared to another algorithm obtained as a direct adaptation of the well-known minimum leakage alternating minimization algorithm for the interference channel, the proposed algorithm is clearly superior in terms of achieved DoF. As a consequence, it appears to be an interesting algorithm not only to find IA solutions but to provide insights into the feasibility of alignment.

6. REFERENCES

[1] S. A. Jafar and S. Shamai (Shitz), "Degrees of freedom region of the MIMO X channel," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 151–170, Jan. 2008.

[2] M. Maddah-Ali, A. Motahari, and A. Khandani, "Communication over MIMO X channels: interference alignment, decomposition, and performance analysis," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3457–3470, Aug. 2008.

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

[4] A. Agustín and J. Vidal, "Degrees of freedom region of the MIMO X channel with an arbitrary number of antennas," *ArXiv preprint available: http://arxiv.org/abs/1210.2582*, Oct. 2012.

[5] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom of wireless X networks," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3893–3908, Sept. 2009.

[6] H. Sun, C. Geng, T. Gou, and S. A. Jafar, "Degrees of freedom of MIMO X networks: spatial scale invariance, one-sided decomposability and linear feasibility," in *2012 IEEE International Symposium on Information Theory Proceedings*, Boston, MA, July 2012, pp. 2082–2086, IEEE.

[7] L. Ke, A. Ramamoorthy, Z. Wang, and H. Yin, "Degrees of freedom region for an interference network with general message demands," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3787–3797, June 2012.

[8] 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, no. 9, pp. 4771–4782, Sept. 2010.

[9] A. Agustín and J. Vidal, "Improved interference alignment precoding for the MIMO X channel," in *IEEE International Conference On Communications (ICC) 2011*, Kyoto, Japan, 2011.

[10] K. S. 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, no. 6, pp. 3309–3322, June 2011.

[11] O. González and I. Santamaría, "Interference alignment in single-beam MIMO networks via homotopy continuation," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2011, pp. 3344–3347, IEEE.

[12] J. R. Magnus and H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics*, John Wiley & Sons, West Sussex, England, third edition, 1999.

[13] H. Sun, C. Geng, T. Gou, and S. A. Jafar, "Degrees of freedom of MIMO X networks: spatial scale invariance, one-sided decomposability and linear feasibility," *ArXiv preprint available: http://arxiv.org/abs/1207.6137*, July 2012.

[14] O. González, C. Beltrán, and I. Santamaría, "A feasibility test for linear interference alignment in MIMO channel with constant coefficients," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1840–1856, Mar. 2014.

[15] T.-Y. Li, "Numerical solution of multivariate polynomial systems," *Acta Numerica*, vol. 6, no. 1, pp. 399–436, 1997.

[16] A. J. Sommese and C. W. Wampler II, *The numerical solution of systems of polynomials arising in engineering and science*, World Scientific Publishing Co. Pte. Ltd., Singapore, Rep. of Singapore, first edition, 2005.

[17] E. L. Allgower and K. Georg, "Numerical path following," in *Handbook of Numerical Analysis*, P. G. Ciarlet and J. L. Lions, Eds., vol. 5, pp. 3–207. Elsevier Inc., Amsterdam, 1997.

[18] A. J. Laub, *Matrix Analysis for Scientists and Engineers*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004.