This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2014.2338132, IEEE Signal Processing Letters

A Quadratically Convergent Method for Interference Alignment in MIMO Interference Channels

Óscar González, *Student Member, IEEE,* Christian Lameiro, *Student Member, IEEE,* and Ignacio Santamaría, *Senior Member, IEEE*

***Abstract*—Alternating minimization and steepest descent are commonly used strategies to obtain interference alignment (IA) solutions in the K-user multiple-input multiple-output (MIMO) interference channel (IC). Although these algorithms are shown to converge monotonically, they experience a poor convergence rate, requiring an enormous amount of iterations which substantially increases with the size of the scenario. To alleviate this drawback, in this letter we resort to the Gauss-Newton (GN) method, which is well-known to experience quadratic convergence when the iterates are sufficiently close to the optimum. We discuss the convergence properties of the proposed GN algorithm and provide several numerical examples showing that it always converges to the optimum with quadratic rate, reducing dramatically the required computation time in comparison to other algorithms, hence paving a new way for the design of IA algorithms.**

***Index Terms*—Interference alignment, alternating minimization, steepest descent, Gauss-Newton, interference channel.**

I. Introduction

Interference alignment (IA) is a promising technique to manage interference in wireless networks [1]. The key idea is to confine the interferences at each receiver into a reduced-dimensional subspace, thus leaving some receiver dimensions free of interference. This approach allows to achieve the maximum degrees-of-freedom (DoF), i.e., the number of data streams that can be transmitted free of interference, which asymptotically characterize the sum-rate capacity.

In this letter we consider the K-user multiple-input multiple-output interference channel (MIMO IC), which is comprised of K transmitter-receiver pairs that interfere with one another. More specifically, we restrict our analysis to the case where no time or frequency symbol extensions are applied and perfect interference alignment is sought (i.e. zero interference leakage). Since closed-form IA solutions are not available for the majority of scenarios, iterative algorithms are usually applied in order to obtain linear precoding and decoding matrices that achieve IA. Over the course of the last years numerous methods have been developed for this task. The first method, which became specially well-known because of its simplicity and reliability, is the alternating minimization algorithm in [2], [3] which we will abbreviate as AltMin in this letter. This algorithm regards the IA problem as an interference leakage minimization, which can be effectively solved by an alternating optimization procedure. It is shown to converge monotonically but does not necessarily achieve a global optimum. Although, to the best of our knowledge, a rigorous proof has not been provided yet, it has been experimentally observed that the AltMin algorithm finds the global solution, attaining zero interference leakage, in all feasible scenarios. The convergence analysis of the AltMin algorithm (as well as all its variants) is typically limited to prove its monotonicity, leaving aside its rate of convergence, which is a key issue for the convergence speed and, consequently, its applicability. As a result, its main drawback is its slow convergence rate, which exacerbates as the problem size (number of users, antennas or streams) increases. Further, AltMin has given rise to many other variants such as [4]–[9] which provide performance improvements at the expense of a higher computational complexity or number of iterations.

Another research line has been that of one-sided algorithms for which the optimization is conducted at either the transmitter or the receiver side of the links. This is the case of the algorithms in [10]–[12] which also resort to an alternating optimization procedure, but pose some issues when dealing with multiple streams and a large number of users. Another promising example are the steepest descent (SD) algorithms [13]–[15] which are guaranteed to converge to a stationary point of the cost function. Many other algorithms with different cost functions (e.g. MMSE, sum-rate) and optimization techniques have been developed [16]–[18], but no clear winner has been found so far, according to recent comparisons [19].

In this letter we propose a method to find IA solutions which does not rely on alternating minimization or steepest descent. We will show that the main limitation of the AltMin and SD algorithms stems from their distributed nature and a completely different approach is needed in pursuit of an improved convergence rate. Our method, instead, is based on the Gauss-Newton (GN) method, which usually shows a quadratic convergence rate. In the case of cost functions on $\mathbb{R}$, the theory behind GN is relatively well-known and understood. Unfortunately, the IA problem poses the substantial difficulty of requiring precoders and decoders to stay full rank along Newton iterations in order to preserve the rank of the desired channels. For the sake of computational savings, we consider in this work a simple but widely used approach, consisting on orthonormalizing the updates at each iteration of the (unconstrained) GN. Given that the updates are small, the orthonormalization step does not

Copyright © 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are with the Department of Communications Engineering, University of Cantabria, 39005 Santander, Spain (e-mail: {oscargf, lameiro, nacho}@gtas.dicom.unican.es) and were supported by the Spanish Government (MICINN) under projects TEC2010-19545-C04-03 (COSIMA), CONSOLIDER-INGENIO 2010 CSD2008-00010 (COMONSENS) and FPU grants AP2009-1105 and AP2010-2189.

1070-9908 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2014.2338132, IEEE Signal Processing Letters

jeopardize the convergence properties of the GN. In this letter, we discuss the convergence properties of the proposed GN and show through several numerical examples that it always converges quadratically to a zero interference leakage point in feasible scenarios. Compared to the well-known AltMin or SD algorithms, our proposed method provides remarkable computation time savings.

II. System model

We consider a K-user MIMO IC comprised of K transmitter-receiver pairs equipped with M_i and N_i antennas, $i = 1,\dots,K$, respectively. Also, each transmitter sends d_i data streams to its corresponding receiver. Using the notation introduced in [20], we expressed this system as $\prod_{i=1}^{K}(M_i \times N_i, d_i)$. The signal at each receiver can be modeled as

$$\mathbf{z}_i = \mathbf{U}_i^H \mathbf{H}_{ii}\mathbf{V}_i\mathbf{s}_i + \sum_{i\neq j}\mathbf{U}_i^H\mathbf{H}_{ij}\mathbf{V}_j\mathbf{s}_j + \mathbf{n}_i \ , \ i = 1,\dots,K \ , \tag{1}$$

where $\mathbf{U}_i \in \mathbb{C}^{N_i\times d_i}$ and $\mathbf{V}_i \in \mathbb{C}^{M_i\times d_i}$ are the decoding and precoding matrices, respectively; $\mathbf{H}_{ij} \in \mathbb{C}^{N_i\times M_j}$ is the flat-fading MIMO channel between transmitter j and receiver i, $\mathbf{s}_i \in \mathbb{C}^{d_i}$ are the symbols transmitted by user i and $\mathbf{n}_i \in \mathbb{C}^{d_i}$ is the additive white Gaussian noise at the ith receiver. 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 IA and the existence of such a precoder design requires the simultaneous satisfiability of the following conditions:

$$\mathbf{U}_i^H\mathbf{H}_{ij}\mathbf{V}_j = \mathbf{0} \ , \ \forall\, i \neq j \ , \tag{2}$$

$$\operatorname{rank}\left(\mathbf{U}_i^H\mathbf{H}_{ii}\mathbf{V}_i\right) = d_i, \quad \forall\, i \ . \tag{3}$$

In fact, condition (3) is almost surely satisfied if the channel matrices $\mathbf{H}_{ij}$ do not have any special structure and both $\mathbf{U}_i$ and $\mathbf{V}_j$ are full column rank [2]. Without loss of generality, it can be assumed that precoders and decoder lie in the Stiefel manifold, i.e. $\mathbf{U}_i^H\mathbf{U}_i = \mathbf{I}_{d_i}$ and $\mathbf{V}_j^H\mathbf{V}_j = \mathbf{I}_{d_j}$ $\forall i,j$.

III. Proposed algorithm

Let us define the vector containing all the optimization variables, that is, the variables in $\mathbf{V}_j$ and $\mathbf{U}_i$ as $\mathbf{x} = [\mathrm{vec}(\mathbf{V}_1)^T,\dots,\mathrm{vec}(\mathbf{V}_K)^T,\mathrm{vec}(\mathbf{U}_1^H)^T,\dots,\mathrm{vec}(\mathbf{U}_K^H)^T]^T$, where $\mathrm{vec}(\mathbf{A})$ denotes the vector obtained by stacking the columns of matrix $\mathbf{A}$ below one another. Consequently, $\mathbf{x}$ contains the totality of $N_v = \sum_i (M_i + N_i)d_i$ variables in the system. Now, we denote as $\mathbf{r}(\mathbf{x})$ the function evaluating the residuals of the equations in (2) which consists of $N_e = \sum_{i\neq j} d_i d_j$ scalar equations, i.e., $\mathbf{r}(\mathbf{x}) = [\mathbf{r}_{21}^T,\dots,\mathbf{r}_{(K-1)K}^T]^T$, where $\mathbf{r}_{ij} = \mathrm{vec}(\mathbf{U}_i^H\mathbf{H}_{ij}\mathbf{V}_j)$. More formally, $\mathbf{r} : \mathbb{C}^{N_v} \to \mathbb{C}^{N_e}$ where $N_v \geq N_e$ is necessary for the system to have a solution. The exact requirements for the system to be feasible have been studied in [21] and references therein, but here we will assume $N_v \geq N_e$ for simplicity. Under these considerations, the interference leakage cost function can be expressed as $f(\mathbf{x}) = \mathbf{r}(\mathbf{x})^H\mathbf{r}(\mathbf{x}) : \mathbb{C}^{N_v} \to \mathbb{R}$.

A. Complex Gauss-Newton method

At the nth iteration of Newton-like methods, the variables are updated according to the rule $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta\mathbf{x}_n$, where the update vector $\Delta\mathbf{x}_n$ is obtained through the second-order approximation of the cost function, $f(\mathbf{x})$. Since $f(\mathbf{x})$ is a real-valued function with complex domain, it is not analytic in $\mathbf{x}$ and hence a Taylor expansion of $f(\mathbf{x})$ at a point $\mathbf{x}_0$ cannot be derived. On the other hand, Wirtinger calculus provides a framework for complex derivation that allows the existence of a complex Taylor expansion of such real-valued functions, by being regarded as a function of the augmented vector $\boldsymbol{\chi} \triangleq [\mathbf{x}^T \ \mathbf{x}^H]^T$. Then, two complex derivatives are defined by taking the derivative with respect to $\mathbf{x}$ while treating $\mathbf{x}^*$ as a constant and the other way around for $\mathbf{x}^*$. For further details, we refer the reader to [22]. Following these lines, the second-order approximation of the interference leakage function, $f(\boldsymbol{\chi})$, around a point $\boldsymbol{\chi}_0$ can be written as [22]:

$$f(\boldsymbol{\chi}) \approx f(\boldsymbol{\chi}_0) + \Delta\boldsymbol{\chi}_0^T\nabla_{\boldsymbol{\chi}}f(\boldsymbol{\chi}_0) + \frac{1}{2}\Delta\boldsymbol{\chi}_0^H\mathcal{H}_{\boldsymbol{\chi}_0}\Delta\boldsymbol{\chi}_0 \ , \tag{4}$$

where $\Delta\boldsymbol{\chi}_0 = \boldsymbol{\chi}-\boldsymbol{\chi}_0$, $\nabla_{\boldsymbol{\chi}}f(\boldsymbol{\chi}_0)$ denotes the complex gradient of the scalar function $f(\boldsymbol{\chi})$ at $\boldsymbol{\chi}_0$ and $\mathcal{H}_{\boldsymbol{\chi}_0}$ denotes the Hessian matrix of $f(\boldsymbol{\chi})$ at $\boldsymbol{\chi}_0$. Note that $f(\boldsymbol{\chi})$ is an alternative representation of $f(\mathbf{x})$ that explicitly shows its dependence on both $\mathbf{x}$ and $\mathbf{x}^*$, and thus $f(\boldsymbol{\chi}) = f(\mathbf{x})$. Let us also denote the Jacobian matrix of the function $\mathbf{r}(\boldsymbol{\chi})$ at $\boldsymbol{\chi} = \boldsymbol{\chi}_0$ by

$$\mathbf{J}_{\boldsymbol{\chi}_0} \triangleq \frac{\partial\mathbf{r}(\boldsymbol{\chi}_0)}{\partial\boldsymbol{\chi}^T} = \begin{bmatrix}\dfrac{\partial\mathbf{r}(\mathbf{x}_0)}{\partial\mathbf{x}^T} & \dfrac{\partial\mathbf{r}(\mathbf{x}_0)}{\partial\mathbf{x}^H}\end{bmatrix} = \begin{bmatrix}\mathbf{J}_{\mathbf{x}_0} & \mathbf{J}_{\mathbf{x}_0^*}\end{bmatrix} \ . \tag{5}$$

Given that $\mathbf{r}(\mathbf{x})$ is analytic in $\mathbf{x}$, i.e., $\mathbf{J}_{\mathbf{x}^*} = \mathbf{0}$, the gradient and the Hessian matrix can be expressed as

$$\nabla_{\boldsymbol{\chi}}f(\boldsymbol{\chi}_0) = \begin{bmatrix}\mathbf{r}(\boldsymbol{\chi}_0)^H\mathbf{J}_{\mathbf{x}_0} & \mathbf{r}(\boldsymbol{\chi}_0)^T\mathbf{J}_{\mathbf{x}_0}^*\end{bmatrix}^T \ , \tag{6}$$

$$\mathcal{H}_{\boldsymbol{\chi}_0} = \begin{pmatrix}\mathring{\mathcal{H}}_{\boldsymbol{\chi}_0} & \tilde{\mathcal{H}}_{\boldsymbol{\chi}_0} \\ \tilde{\mathcal{H}}_{\boldsymbol{\chi}_0}^* & \mathring{\mathcal{H}}_{\boldsymbol{\chi}_0}^*\end{pmatrix} \ , \tag{7}$$

where

$$\mathring{\mathcal{H}}_{\boldsymbol{\chi}_0} = \mathbf{J}_{\mathbf{x}_0}^H\mathbf{J}_{\mathbf{x}_0} + \underbrace{\sum_k r_k(\mathbf{x}_0)\frac{\partial^2 r_k^*(\mathbf{x}_0)}{\partial\mathbf{x}^T\partial\mathbf{x}^*}}_{=0} \ , \tag{8}$$

$$\tilde{\mathcal{H}}_{\boldsymbol{\chi}_0} = \underbrace{\mathbf{J}_{\mathbf{x}_0^*}^H\mathbf{J}_{\mathbf{x}_0^*}}_{=0} + \sum_k r_k(\mathbf{x}_0)\frac{\partial^2 r_k^*(\mathbf{x}_0)}{\partial\mathbf{x}^H\partial\mathbf{x}^*} \ , \tag{9}$$

are the complex and complementary Hessian matrices, respectively, and r_k denotes the kth element of $\mathbf{r}$.

In the GN method, the Hessian matrix is approximated by taking $\tilde{\mathcal{H}}_{\boldsymbol{\chi}} = \mathbf{0}$, which is a reasonable approximation when the entries of $\mathbf{r}$ are small (we are close to the minimum) or the function $\mathbf{r}$ is mildly non-linear (the second derivatives are small). As $\mathbf{r}$ is a bilinear function, this happens to be a rather good approximation. Taking this approximation into account, and using (6)–(9), we can express (4) as a function of $\mathbf{x}$ as

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + 2\Re\left\{\mathbf{r}(\mathbf{x}_0)^H\mathbf{J}_{\mathbf{x}_0}\Delta\mathbf{x}_0\right\} + \Delta\mathbf{x}_0^H\mathbf{J}_{\mathbf{x}_0}^H\mathbf{J}_{\mathbf{x}_0}\Delta\mathbf{x}_0 \ . \tag{10}$$

Note that the approximated Hessian is positive semidefinite, and thus (10) is actually a convex approximation of $f(\mathbf{x})$ at

1070-9908 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2014.2338132, IEEE Signal Processing Letters

$\mathbf{x}_0$. Finally, the GN update is obtained when the derivative of (10) with respect to $\Delta\mathbf{x}_0$ equals zero:

$$\frac{\partial f(\mathbf{x})}{\partial \Delta\mathbf{x}_0} = 2\mathbf{J}_{\mathbf{x}_0}^H \mathbf{r}(\mathbf{x}_0) + 2\mathbf{J}_{\mathbf{x}_0}^H \mathbf{J}_{\mathbf{x}_0}\Delta\mathbf{x}_0 = \mathbf{0}. \quad (11)$$

Given that $\mathbf{J}_{\mathbf{x}_0}$ is $N_e \times N_v$ and $N_v \geq N_e$, (11) simplifies to $\mathbf{J}_{\mathbf{x}_0}\Delta\mathbf{x}_0 = -\mathbf{r}(\mathbf{x}_0)$, whose infinite solutions can be parametrized as $\Delta\mathbf{x}_0 = -\mathbf{J}_{\mathbf{x}_0}^\dagger \mathbf{r}(\mathbf{x}_0) + (\mathbf{I} - \mathbf{J}_{\mathbf{x}_0}^\dagger \mathbf{J}_{\mathbf{x}_0})\mathbf{w}$, where $(\cdot)^\dagger$ denotes the Moore-Penrose (MP) pseudoinverse and $\mathbf{w}$ is a vector of free parameters. Among all solutions, it is reasonable to pick the one that is normal to the manifold $\{\mathbf{x} : \mathbf{r}(\mathbf{x}) = \mathbf{r}(\mathbf{x}_0)\}$ (the set of variables keeping residuals unchanged) or, equivalently, is orthogonal to the nullspace of $\mathbf{J}_{\mathbf{x}_0}$. More formally, this is achieved by setting $(\mathbf{I} - \mathbf{J}_{\mathbf{x}_0}^\dagger \mathbf{J}_{\mathbf{x}_0})\mathbf{w} = \mathbf{0}$ which leads to the so-called *normal flow update*, i.e. $\Delta\mathbf{x}_0 = -\mathbf{J}_{\mathbf{x}_0}^\dagger \mathbf{r}(\mathbf{x}_0)$ [23]. In practice, it is recommended not to compute the MP pseudoinverse but, instead, solve

$$\operatorname{argmin}_{\Delta\mathbf{x}_n}\{\|\Delta\mathbf{x}_n\| : \mathbf{J}_{\mathbf{x}_n}\Delta\mathbf{x}_n = -\mathbf{r}(\mathbf{x}_n)\}\ . \quad (12)$$

Most of the existing linear algebra routines for solving this problem cannot exploit the sparse structure of $\mathbf{J}_{\mathbf{x}_n}$ or, if they do, compute a fast *basic* solution instead of the minimum-norm solution. A convenient routine fulfilling both requirements is `SPQR_SOLVE` which is part of the SuiteSparseQR linear algebra bundle by Davis [24].

Finally, it is worth pointing out that due to the fact the GN updates are small, the precoders and decoders obtained after each iteration should guarantee condition (3). Therefore, both precoders and decoders can be projected back to the Stiefel manifold by computing an orthonormal basis of the subspace spanned by each of them. Given that the interference leakage function is invariant in the Grassmann manifold, the particular choice of orthonormal representatives is irrelevant. Therefore, among all orthonormalization operations, we consider the QR decomposition as it requires the least computational demands, and denote the $\mathbf{Q}$ factor as $\mathrm{qf}(\cdot)$. The complete procedure is summarized in Algorithm 1.

Remark: The GN method applied to minimize $f(\mathbf{x}) = \mathbf{r}(\mathbf{x})^H\mathbf{r}(\mathbf{x})$ is identical to the classical Newton's method applied to the system of equations $\mathbf{r}(\mathbf{x}) = \mathbf{0}$ when the minimum norm update is chosen [25]. To see this, consider a first order model of $\mathbf{r}(\mathbf{x})$, i.e. $\mathbf{r}(\mathbf{x}) = \mathbf{r}(\mathbf{x}_0) + \mathbf{J}_{\mathbf{x}_0}\Delta\mathbf{x}_0$. In the classical Newton's method, the update vector $\Delta\mathbf{x}_0$ must satisfy $\mathbf{r}(\mathbf{x}_0) + \mathbf{J}_{\mathbf{x}_0}\Delta\mathbf{x}_0 = \mathbf{0}$ thus yielding the update in (12).

B. Some remarks on the convergence properties

Convergence of GN methods is usually difficult to analyze, hence we provide here some insights based on empirical observations rather than formal convergence proofs. Nevertheless, we will observe in Section V through exhaustive simulations that our intuitions behind the convergence of the method are in agreement with the experimental results.

1) Stationary points: From (11) it is clear that points satisfying $\mathbf{J}_{\mathbf{x}_0}^H\mathbf{r}(\mathbf{x}_0) = \mathbf{0}$ are accumulation points, i.e., stationary points of the method. In a previous work [21] we have proved that, for a feasible IA system, the matrix $\mathbf{J}_{\mathbf{x}}$ is always full-rank, and therefore the nullspace of $\mathbf{J}_{\mathbf{x}}^H$ is always empty. Thus, these points correspond to $\mathbf{r}(\mathbf{x}) = \mathbf{0}$, i.e., zero interference leakage. Also, points at which the updates do not change the subspace of the precoders and decoders are also accumulation points (recall that the interference leakage is invariant in the Grassmann manifold), but do not necessarily correspond to stationary points of the interference leakage. Note, however, that such points are also present in the AltMin and other IA algorithms.

Choose a tolerance level, δ, and the initial point $\{\mathbf{V}_{i,0}, \mathbf{U}_{i,0}\}_{i=1}^K$, lying on the Stiefel manifold.
Set $n = 0$.
repeat
1) Construct $\mathbf{x}_n$, $\mathbf{J}_{\mathbf{x}_n}$ and $\mathbf{r}(\mathbf{x}_n)$ and solve (12) for $\Delta\mathbf{x}_n$.
2) Construct $\Delta\mathbf{V}_{i,n}$ and $\Delta\mathbf{U}_{i,n}$ from $\Delta\mathbf{x}_n$ and compute

$$\mathbf{V}_{i,n+1} = \mathrm{qf}(\mathbf{V}_{i,n} + \Delta\mathbf{V}_{i,n})\ ,$$
$$\mathbf{U}_{i,n+1} = \mathrm{qf}(\mathbf{U}_{i,n} + \Delta\mathbf{U}_{i,n})\ .$$

3) $n = n + 1$.
until $f(\mathbf{x}_n) \leq \delta$.

Algorithm 1: GN method for interference leakage minimization.

2) Non-monotone convergence: It can be seen that the classical GN direction is a descent direction of the function $f(\mathbf{x})$ when $\mathbf{J}_{\mathbf{x}}^H\mathbf{r}(\mathbf{x})$ is nonzero [25]. In other words, the scalar product of the direction $\Delta\mathbf{x}$ over the gradient is always negative, i.e. $\mathbf{r}(\mathbf{x})^H\mathbf{J}_{\mathbf{x}}\Delta\mathbf{x} < 0$. Intuitively, it is clear that the interference leakage can always be reduced by diminishing the transmitted power, thus guaranteeing a monotone convergence. In general, when a power constraint is added (e.g. by restricting precoders and decoders to lie in the Stiefel manifold as in Step 2 of Algorithm 1) monotone convergence does not hold anymore.

IV. Rate of convergence

A sequence of vectors $\{\mathbf{x}_n\}$ is said to converge to $\mathbf{x}^\star$ with order α if

$$\lim_{n\to\infty} \frac{\|\mathbf{x}_{n+1} - \mathbf{x}^\star\|}{\|\mathbf{x}_n - \mathbf{x}^\star\|^\alpha} = c\ , \quad (13)$$

with $0 \leq c < \infty$. For example, the classical GN method is known to converge q-quadratically (i.e. $\alpha = 2$) for small residual problems, $\mathbf{r}(\mathbf{x}^\star) \simeq \mathbf{0}$, when the following assumptions are satisfied [25]: the residuals $r_k(\mathbf{x})$ are Lipschitz continuously differentiable (i.e., their second derivative is bounded) and the Jacobian $\mathbf{J}_{\mathbf{x}}$ is full rank for all $\mathbf{x}$ in a neighborhood of the optimum $\mathbf{x}^\star$. Since both requirements are met in the IA problem (recall that the Jacobian matrix is always full-rank for feasible scenarios), GN is expected to converge q-quadratically in a neighborhood of the optimum. We note that q-quadratic convergence holds for the classical GN method but may not hold when additional operations such as the orthonormalization in Step 2 of Algorithm 1 are applied. Fortunately, in a neighborhood of the optimum, the orthonormalization step can be regarded as a *retraction* which guarantees superlinear convergence ($\alpha > 1$) [26]. Our numerical results suggest the

1070-9908 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2014.2338132, IEEE Signal Processing Letters

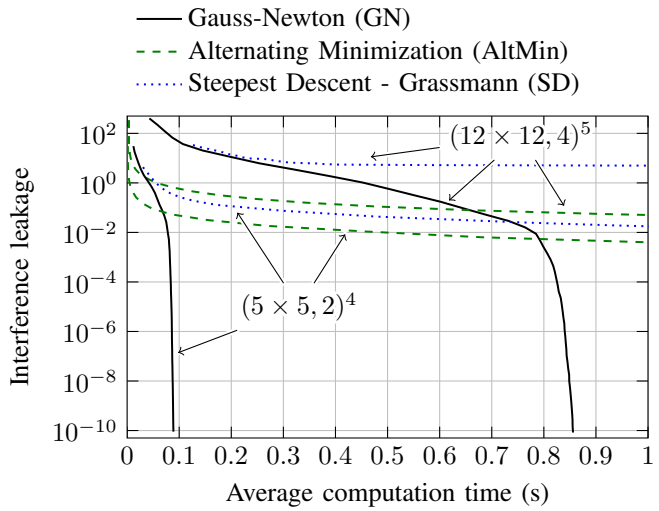

Fig. 1. Average convergence of GN, AltMin and SD for the $(5 \times 5, 2)^4$ and $(12 \times 12, 4)^5$ scenarios.

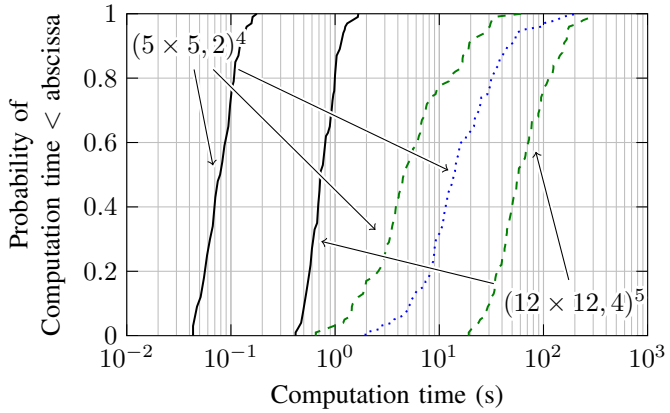

Fig. 2. CDF of computation times of the GN, AltMin and SD algorithms in scenarios $(5 \times 5, 2)^4$ and $(12 \times 12, 4)^5$.

TABLE I
MEDIAN NUMBER OF ITERATIONS TO REACH AN INTERFERENCE LEAKAGE OF 10^{-5} AND AVERAGE TIME PER ITERATION.

Scenario	Median number of iterations			Iteration time (ms)		
	GN	AltMin	SD	GN	AltMin	SD
$(5 \times 5, 2)^4$	20	6204	1917	3.9	0.7	7.3
$(12 \times 12, 4)^5$	42	32190	–	18.5	1.8	28.0

convergence rate is indeed q-quadratic although a rigorous proof is not available so far.

On the other hand, both alternating-optimization and steepest descent algorithms on manifolds are known to converge q-linearly (that is, $\alpha = 1$) with $0 \leq c < 1$ (see [27] and [26], respectively). These algorithms, despite being simple, lack good rate of convergence properties, making them prohibitively slow. This limitation stems from their distributed nature, constraining the optimization problem to a subset of variables at each iteration. Conversely, the GN method takes advantage of a joint, centralized optimization which enables a more focused convergence.

V. NUMERICAL RESULTS

In this section we provide several numerical examples to compare the convergence speed of the proposed GN method to that of the AltMin [2] and SD [15] algorithms. Our results are averaged over 100 independent Monte-Carlo simulations, where the entries of the MIMO channels are independent and identically distributed complex Gaussian variables with zero mean and unit variance.

The evolution of the interference leakage with the average computation time (in an Intel i7 3.2 GHz CPU) for the scenarios $(5 \times 5, 2)^4$ and $(12 \times 12, 4)^5$ is depicted in Fig. 1. The difference in the convergence rate between AltMin and SD on the one hand and the GN method on the other is readily observed. More specifically, the use of AltMin against GN would be only justified when the desired interference level is still far above 10^{-2}, which is not a sufficiently low value for the signal-to-noise ratio (SNR) regimes where IA is meaningful. For the considered scenarios, the SD algorithm is always slower than AltMin and, in fact, fails to converge (stagnating in local minima) in the $(12 \times 12, 4)^5$ scenario. On the other hand, both AltMin and GN have always converged to a zero-leakage solution. The CDF of the computation times and the median number of iterations to reach an interference leakage of 10^{-5} are depicted in Fig. 2 and Table I, respectively.

Lastly, we analyze the convergence order of the two algorithms that have always converged in both scenarios: GN and AltMin. We estimate the convergence order, α, by means of the formula

$$\alpha \approx \frac{\log(\|\mathbf{x}_{n+1} - \mathbf{x}_n\| \,/\, \|\mathbf{x}_n - \mathbf{x}_{n-1}\|)}{\log(\|\mathbf{x}_n - \mathbf{x}_{n-1}\| \,/\, \|\mathbf{x}_{n-1} - \mathbf{x}_{n-2}\|)}.$$

The GN method gives $\alpha = 2.10$ and $\alpha = 2.05$ for the scenarios $(5 \times 5, 2)^4$ and $(12 \times 12, 4)^5$, respectively, thus showing that the convergence is q-quadratic and corroborating our arguments in Sections III-B and IV. The estimates of α for the AltMin algorithm are $\alpha = 0.91$ and $\alpha = 1.01$, respectively, which are also consistent with the q-linear convergence results in the literature.

VI. CONCLUSION

In this letter we have proposed a new algorithm for the IA problem in the K-user MIMO IC. The proposed algorithm is based upon the Gauss-Newton method, which is well-known for its quadratic convergence rate. We have discussed the convergence properties of the proposed approach, which have been validated through exhaustive numerical simulations, showing that the proposed algorithm does always converge to the optimal solution and does it quadratically. Consequently, the computation time is dramatically reduced in comparison to steepest descent and the widely-used alternating minimization algorithm. These findings pose a novel approach to obtain IA solutions, endowed with a remarkable speed of convergence, which computes (even for very complex scenarios) IA solutions in a fraction of the time required by any other state-of-the-art algorithm.

1070-9908 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2014.2338132, IEEE Signal Processing Letters

REFERENCES

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

[2] K. Gomadam, V. Cadambe, and S. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3309–3322, June 2011.

[3] S. W. Peters and R. W. Heath Jr., "Interference Alignment via Alternating Minimization," in *Proc. IEEE ICASSP*, Taipei, Taiwan, 2009, pp. 2446–2448.

[4] C. Lameiro, Ó. González, and I. Santamaría, "An interference alignment algorithm for structured channels," in *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2013)*, Darmstadt, Germany, Jun. 2013.

[5] I. Santamaría, Ó. González, R. W. Heath Jr., and S. W. Peters, "Maximum sum-rate interference alignment algorithms for MIMO channels," in *Proc. IEEE Globecom*, Miami, FL, USA, 2010.

[6] P. Mohapatra, K. E. Nissar, and C. R. Murthy, "Interference Alignment Algorithms for the K User Constant MIMO Interference Channel," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5499–5508, Nov. 2011.

[7] D. S. Papailiopoulos and A. G. Dimakis, "Interference Alignment as a Rank Constrained Rank Minimization," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4278–4288, Aug. 2012.

[8] G. C. Alexandropoulos and C. B. Papadias, "A reconfigurable iterative algorithm for the K-user MIMO interference channel," *Signal Processing*, vol. 93, no. 12, pp. 3353–3362, Jun. 2013.

[9] K. Raj Kumar and F. Xue, "An iterative algorithm for joint signal and interference alignment," in *2010 IEEE International Symposium on Information Theory*, Jun. 2010, pp. 2293–2297.

[10] H. Yu and Y. Sung, "Least Squares Approach to Joint Beam Design for Interference Alignment in Multiuser Multi-Input Multi-Output Interference Channels," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4960–4966, Sep. 2010.

[11] H. Yu, Y. Sung, H. Kim, and Y. H. Lee, "Beam Tracking for Interference Alignment in Slowly Fading MIMO Interference Channels: A Perturbations Approach Under a Linear Framework," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1910–1926, Apr. 2012.

[12] J. Lee, H. Yu, and Y. Sung, "Beam Tracking for Interference Alignment in Time-Varying MIMO Interference Channels: A Conjugate Gradient Based Approach," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 958–964, 2013.

[13] H. G. Ghauch and C. B. Papadias, "Interference Alignment: A One-Sided Approach," in *2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, Houston, TX, USA, Dec. 2011, pp. 1–5.

[14] S. Bazzi, G. Dietl, and W. Utschick, "Interference alignment via minimizing projector distances of interfering subspaces," in *2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Cesme, Turkey, Jun. 2012, pp. 274–278.

[15] C. Zhang, H. Yin, and G. Wei, "One-Sided Precoder Designs for Interference Alignment," in *2012 IEEE Vehicular Technology Conference (VTC 2012 Fall)*, Québec City, Canada, Sep. 2012.

[16] 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 Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[17] B. Zhu, J. Ge, J. Li, and C. Sun, "Subspace optimisation-based iterative interference alignment algorithm on the Grassmann manifold," *IET Communications*, vol. 6, no. 18, pp. 3084–3090, Dec. 2012.

[18] Y. Zhao, S. N. Diggavi, A. Goldsmith, and H. V. Poor, "Convex optimization for precoder design in MIMO interference networks," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, Oct. 2012, pp. 1213–1219.

[19] D. Schmidt, C. Shi, R. Berry, M. Honig, and W. Utschick, "Comparison of Distributed Beamforming Algorithms for MIMO Interference Networks," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3476–3489, 2013.

[20] C. Yetis, T. Gou, S. Jafar, and A. Kayran, "On feasibility of interference alignment in MIMO interference networks," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4771–4782, Sep. 2010.

[21] Ó. González, C. Beltrán, and I. Santamaría, "A feasibility test for linear interference alignment in MIMO channels with constant coefficients," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1840–1856, Mar. 2014.

[22] P. Schreier and L. Scharf, *Statistical Signal Processing of Complex-Valued Data: The Theory of Improper and Noncircular Signals*. Cambridge University Press, 2010.

[23] H. F. Walker and L. T. Watson, "Least-Change Secant Update Methods for Underdetermined Systems," *SIAM Journal on Numerical Analysis*, vol. 27, no. 5, pp. 1227–1262, Aug. 1990.

[24] T. A. Davis, "Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization," *ACM Trans. Math. Software*, vol. 38, no. 1, pp. 8:1–8:22, Dec. 2011.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[26] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, New Jersey: Princeton University Press, 2008.

[27] J. Bezdek and R. Hathaway, "Some notes on alternating optimization," in *Advances in Soft Computing AFSS 2002*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, vol. 2275, pp. 288–300.

1070-9908 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.