One-sided Precoder Designs for Interference Alignment

Chen Zhang, Huarui Yin, and Guo Wei
Department of Electrical Engineering and Information Science
University of Science and Technology of China, HeFei, Anhui, 230027, P.R.China
Email: zhangzc@mail.ustc.edu.cn, {yhr, wei}@ustc.edu.cn

***Abstract*—The aim of this paper is to propose a fast convergence algorithm of precoder to achieve feasible interference alignment. By limiting the optimization only on the transmitters' side, it relaxes the assumption of channel reciprocity which will alleviate the significant overhead induced by alternating between the forward and reverse communication links. A lower complexity and higher robustness modified steepest descent (SD) algorithm in the complex space is introduced first. Then we reform the optimization problem on the complex Stiefel manifold and derive a novel SD algorithm to achieve perfect interference alignment. Simulation results suggest that comparing with previous methods, the novel SD algorithm on Stiefel manifold has better convergence performance and higher system capacity.**

I. Introduction

Interference alignment (IA) is a technique recently proposed in [1] and [2], which can achieve much higher wireless networks capacity than previously believed [3]. For a K-user $M \times M$ MIMO Interference Channel, the sum capacity is

$$C_{sum} = \frac{KM}{2}\log(1+SNR) + o(\log(SNR)), \qquad (1)$$

with achievability of $KM/2$ degrees of freedom (DoF), which is defined as

$$DoF = \lim_{SNR\to\infty} \frac{C_{sum}}{\log(SNR)} \qquad (2)$$

The main idea of IA is to coordinate transmitting directions in order to force the interference overlapping at each receiver, making half of the signaling space interference-free for the desired signals. Therefore, irrespective of the number of interferers, each transmitter-receiver pair is able to communicate with achievability DoF of $M/2$.

An iterative algorithm that utilizes the channel reciprocity to achieve interference alignment, by alternating between the forward and reverse communication links in a distributed way, was proposed in [4]. However, with the assumption of channel reciprocity, the applicability of the algorithm is limited to TDD systems only. Moreover, alternation between the forward and reverse links needs tight synchronization at each node, which may induce too much overhead when the channel varies quickly. Furthermore, the transmitters and receivers exchange their "roles" during each iteration of optimization, which is inappropriate for the receivers having limited computing capability, for example, the downlink of cellular system. Similar to [4], some previous work such as [5] and [6] achieve feasible interference alignment by iteratively optimizing both the precoder matrices and interference suppression filters, which means both the transmitters and receivers are active in the iteration. Obviously the drawbacks we previously encountered will be the bottleneck of application in [5] and [6].

In this paper, we offer a strategy to limit the optimization only on the transmitters' side. For the same purpose, an intuitive steepest descent (SD) algorithm is proposed in [7], which, however defines inner product and gradient direction in inappropriate topologies. At first we modify [7] by rectifying the flaw and reducing the computation complexity. Then we reformulate the constrained optimization problem to an unconstrained one on the complex Stiefel manifold, and break the tradition in [8] by not moving descent step along the geodesic which is the generalization of a straight line to constrained surface [9]. Thus a novel steepest descent algorithm on Stiefel manifold is proposed to achieve feasible interference alignment. In respect of receivers, the proposed algorithm is transparent because only transmitters participate in the iteration, which means it bypasses the overhead and other complications generated by alternation between the forward and reverse network. Moreover, by relaxing the assumption of channel reciprocity, our algorithm is applicable to both TDD and FDD systems. Furthermore, numerical simulation shows that the novel algorithm has better convergence performance and higher system capacity. Finally, we prove that the algorithms converge with respect to our cost function.

Notation: We use bold uppercase letters for matrices or vectors. $\boldsymbol{X}^T$ and $\boldsymbol{X}^\dagger$ denote the transpose and the conjugate transpose (Hermitian) of the matrix $\boldsymbol{X}$ respectively. Assuming the eigenvalues of a matrix $\boldsymbol{X}$ and their corresponding eigenvectors are sorted in ascending order, λ_X^i denotes the i^{th} eigenvalue of the matrix $\boldsymbol{X}$. Then $\boldsymbol{I}$ represents the identity matrix. Moreover $tr(\cdot)$ indicates the trace, and the Euclidean norm of $\boldsymbol{X}$ is $\|\boldsymbol{X}\| = \sqrt{tr(\boldsymbol{X}^\dagger\boldsymbol{X})}$. $vec(\boldsymbol{X})$ represents the staked columns of $\boldsymbol{X}$. $\mathbb{C}^{n\times p}$ represents the $n\times p$ dimensional complex space assuming $n > p$. And $\mathbb{R}^+$ represents the positive real number space. $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and imaginary parts of a complex quantity, respectively. Finally $\kappa = \{1, ..., K\}$ is the set of integers from 1 to K.

II. System model

Consider the K-user wireless MIMO interference channel depicted in Fig.1 where each transmitter and receiver are

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

equipped with $M^{[k]}$ and $N^{[k]}$ antennas respectively. Each transmitter communicates with its corresponding receiver, and creates interference to all the other receivers. $d^{[k]}$ is the desired number of data streams between the k^{th} transmitter-receiver pair. Additionally, $\boldsymbol{H}^{[kj]}$ denotes the $N^{[k]} \times M^{[j]}$ channel coefficients matrix from the j^{th} transmitter to the k^{th} receiver, and is assumed to have i.i.d. complex Gaussian random variables, drawn from a continuous distribution. Finally the received signal vector at receiver k after zero-forcing the interference is denoted by

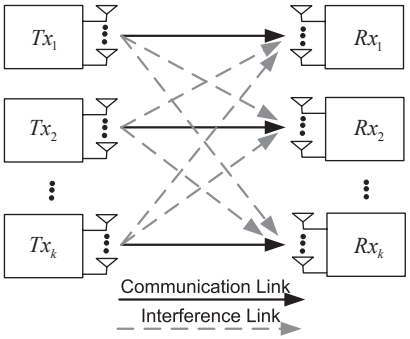

Fig. 1. K-user MIMO interference channel

$$\overline{\boldsymbol{Y}}^{[k]} = \boldsymbol{U}^{[k]\dagger}\boldsymbol{Y}^{[k]} = \boldsymbol{U}^{[k]\dagger}(\sum_{j=1}^{K} \boldsymbol{H}^{[kj]}\boldsymbol{V}^{[j]}\boldsymbol{S}^{[j]} + \boldsymbol{W}^{[k]}), k \in \kappa, \tag{3}$$

where each element of the $d^{[j]} \times 1$ vector $\boldsymbol{S}^{[j]}$ represents an independently encoded Gaussian symbol with power $P^{[j]}/d^{[j]}$ that beamformed with the corresponding $M^{[j]} \times d^{[j]}$ precoder matrix $\boldsymbol{V}^{[j]}$, and then transmitted by the transmitter j. $\boldsymbol{U}^{[k]}$ is the $N^{[k]} \times d^{[k]}$ interference zero-forcing filter at the receiver k. And $\boldsymbol{W}^{[k]}$ is the i.i.d. complex Gaussian noise with zero mean unite variance.

III. Algorithm

A. Feasibility of Interference Alignment

The quality of alignment is measured by the interference power remaining in the intended signal subspace at each receiver. Therefore interference alignment can be achieved by progressively reducing the leakage interference. And if interference alignment is feasible, the leakage interference can eventually be coordinated to zero. From [4], it can be obtained that the $d^{[k]}$ dimensional received signal subspace that contains the least interference is the space spanned by the eigenvectors corresponding to the $d^{[k]}$ smallest eigenvalues of the interference covariance matrix $Q^{[k]}$. Above all, we try to minimize the sum of interference power spilled to the desired signal subspaces, by minimizing the sum of the $d^{[k]}$-smallest eigenvalues of the interference covariance matrix at each receiver and to create $d^{[k]}$-dimensional interference-free subspace for the desired signal.

B. Mathematical Model

As previously stated, we try to minimize the sum of the $d^{[k]}$-smallest eigenvalues (in absolute value) of the interference covariance matrix at each receiver, over the set of precoder matrices $\boldsymbol{V}^{[1]}, ..., \boldsymbol{V}^{[K]}$. Therefore, we define the cost function [7] by:

$$\begin{aligned} &\min_{\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[K]}} f = \sum_{k=1}^{K} \sum_{i=1}^{d^{[k]}} \left| \lambda^{i}_{Q^{[k]}} \right|, \ k, j \in \kappa \\ &\text{subject to } \boldsymbol{V}^{[j]\dagger}\boldsymbol{V}^{[j]} = \boldsymbol{I}_{d^{[j]}} \end{aligned} \tag{4}$$

where

$$\boldsymbol{Q}^{[k]} = \sum_{\substack{j=1 \\ j \neq k}}^{K} \frac{P^{[j]}}{d^{[j]}} \boldsymbol{H}^{[kj]}\boldsymbol{V}^{[j]}\boldsymbol{V}^{[j]\dagger}\boldsymbol{H}^{[kj]\dagger} \tag{5}$$

is the interference covariance matrix at receiver k. With the assumption that all the eigenvalues are sorted in ascending order, $\lambda^{i}_{Q^{[k]}}$ represents the i^{th} eigenvalue of the corresponding interference covariance matrix $\boldsymbol{Q}^{[k]}$. And because $\boldsymbol{Q}^{[k]}$ is a Hermitian matrix, all its eigenvalues are real. Finally, the cost function $f(\boldsymbol{V})$, $f : \mathbb{C}^{n \times p} \rightarrow \mathbb{R}^{+}$ is built.

C. The Steepest Descent Algorithm in Complex Space

Intuitively, the steepest descent method goes hand-in-hand with derivative and differentiation. In order to get the derivative of $f(\boldsymbol{V})$ over $\boldsymbol{V}$, four Jacobian matrices blocks are computed in [7]. Our proposed algorithm reduces the computation complexity by using only two Jacobian matrices blocks:

$$df = \begin{bmatrix} \boldsymbol{D}_R^{[1]} & ... & \boldsymbol{D}_R^{[K]} \end{bmatrix} \begin{bmatrix} d\boldsymbol{V}_R^{[1]} \\ \cdot \\ \cdot \\ d\boldsymbol{V}_R^{[K]} \end{bmatrix} + \begin{bmatrix} \boldsymbol{D}_I^{[1]} & ... & \boldsymbol{D}_I^{[K]} \end{bmatrix} \begin{bmatrix} d\boldsymbol{V}_I^{[1]} \\ \cdot \\ \cdot \\ d\boldsymbol{V}_I^{[K]} \end{bmatrix} \tag{6}$$

where $\boldsymbol{V}_R^{[j]} = \Re\{\boldsymbol{V}^{[j]}\}$, and $\boldsymbol{V}_I^{[j]} = \Im\{\boldsymbol{V}^{[j]}\}$. $\boldsymbol{D}_R^{[j]}$ and $\boldsymbol{D}_I^{[j]}$ are the $d^{[j]} \times M^{[j]}$ Jacobian matrices which denote the partial differential relation of the cost function over the real and imaginary parts of $\boldsymbol{V}^{[j]}$ respectively. The detail of mathematical derivations can be found in [7] and [10].

In [7] the inner product and the gradient direction are defined in different topologies. It is considered to be inappropriate because the gradient is defined after the inner product is given only. In other words, the inner product and the gradient direction must be defined in the same topology. We rectify the flaw and present our modified algorithm in the table below. Inspired by [9], the structure of the proposed algorithm is intuitive. Here some explanations are presented. In Step 3 and Step 4, the Armijo step rule [11] is performed to find a proper convergence step length. Generally speaking, Step 3 ensures the chosen step $\beta^{[j]}$ will significantly reduce the cost function while Step 4 prevents $\beta^{[j]}$ from being too large that may miss the potential optimal point. The operator $gs(\cdot)$ means Gram-Schmidt Orthogonalization [10] of a matrix, which guarantees

the newly computed solution $\boldsymbol{V}^{[j]}$ (or $\boldsymbol{B}_1^{[j]}$, $\boldsymbol{B}_2^{[j]}$) still satisfies the unitary constraint.

Algorithm 1 The Steepest Descent Algorithm in Complex Space

Start with arbitrary precoder matrices $\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[K]}$, $\boldsymbol{V}^{[j]\dagger}\boldsymbol{V}^{[j]} = \boldsymbol{I}$ and begin iteration.

for $j = 1...K$

(1) Compute the Jacobian matrix $\boldsymbol{D}_R^{[j]}$ and $\boldsymbol{D}_I^{[j]}$

(2) Get the steepest descent direction

$$\boldsymbol{Z}^{[j]} = -\boldsymbol{D}_V^{[j]} = -(\boldsymbol{D}_R^{[j]} + i\boldsymbol{D}_I^{[j]})^T$$

(3) Compute $\boldsymbol{B}_1^{[j]} = gs(\boldsymbol{V}^{[j]} + 2\beta^{[j]}\boldsymbol{Z}^{[j]})$,
if $f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{K}) - f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[j-1]},\boldsymbol{B}_1^{[j]},\boldsymbol{V}^{[j+1]}...\boldsymbol{V}^{[K]})$ $\geq \beta^{[j]}tr(\boldsymbol{Z}^{[j]\dagger}\boldsymbol{Z}^{[j]})$, then set $\beta^{[j]} := 2\beta^{[j]}$, repeat Step 3.

(4) Compute $\boldsymbol{B}_2^{[j]} = gs(\boldsymbol{V}^{[j]} + \beta^{[j]}\boldsymbol{Z}^{[j]})$,
if $f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[K]}) - f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[j-1]},\boldsymbol{B}_2^{[j]},\boldsymbol{V}^{[j+1]}...\boldsymbol{V}^{[K]})$ $< \frac{1}{2}\beta^{[j]}tr(\boldsymbol{Z}^{[j]\dagger}\boldsymbol{Z}^{[j]})$, then set $\beta^{[j]} := \frac{1}{2}\beta^{[j]}$, repeat Step 4.

(5)$\boldsymbol{V}^{[j]} = gs(\boldsymbol{V}^{[j]} + \beta^{[j]}\boldsymbol{Z}^{[j]})$

(6) Continue till the cost function f is sufficiently small.

D. The Steepest Descent Algorithm on Complex Stiefel Manifold

The constraint condition $\boldsymbol{V}^{[j]\dagger}\boldsymbol{V}^{[j]} = \boldsymbol{I}$, inspires us to solve the problem on the complex Stiefel manifold. Firstly, some definitions and derivations about the complex Stiefel manifold are introduced. The complex Stiefel manifold [10] $St(n,p)$ is the set

$$St(n,p) = \{\boldsymbol{X} \in \mathbb{C}^{n\times p} : \boldsymbol{X}^\dagger\boldsymbol{X} = \boldsymbol{I}\}. \tag{7}$$

$St(n,p)$ naturally embeds in $\mathbb{C}^{n\times p}$ and inherits the usual topology of $\mathbb{C}^{n\times p}$. Another important definition is the projection. Assuming $\boldsymbol{Y} \in \mathbb{C}^{n\times p}$ is a rank p matrix, the projection operator π is given by

$$\pi(\boldsymbol{Y}) = \arg\min_{\boldsymbol{X}\in St(n,p)} \|\boldsymbol{Y} - \boldsymbol{X}\|^2. \tag{8}$$

From (8), it can be acquired that the projection of an arbitrary rank p matrix $\boldsymbol{Y}$ onto the Stiefel manifold is defined to be the point on the Stiefel manifold closest to $\boldsymbol{Y}$ in the Euclidean norm [9]. Besides, if the SVD of $\boldsymbol{Y}$ is $\boldsymbol{Y} = \boldsymbol{U}\sum\boldsymbol{V}^\dagger$,

$$\pi(\boldsymbol{Y}) = \boldsymbol{U}\boldsymbol{I}_{n\times p}\boldsymbol{V}^\dagger. \tag{9}$$

The tangent space $T_X(n,p)$ at $\boldsymbol{X} \in St(n,p)$ of the Stiefel manifold is defined by

$$T_X(n,p) = \{\boldsymbol{Z} \in \mathbb{C}^{n\times p} : \boldsymbol{Z} = \boldsymbol{X}\boldsymbol{A} + \boldsymbol{X}_\perp\boldsymbol{B}, \boldsymbol{A} \in \mathbb{C}^{p\times p}, \boldsymbol{A} + \boldsymbol{A}^\dagger = 0, \boldsymbol{B} \in \mathbb{C}^{(n-p)\times p}\}, \tag{10}$$

where $\boldsymbol{X}_\perp \in \mathbb{C}^{n\times(n-p)}$ is defined to be any matrix satisfying $[\boldsymbol{X}\ \boldsymbol{X}_\perp]^\dagger[\boldsymbol{X}\ \boldsymbol{X}_\perp] = \boldsymbol{I}$ and is the complement of $\boldsymbol{X} \in St(n,p)$. Also from [9], it can be obtained that, the gradient matrices of our cost function are in the tangent space when we consider the problem on Stiefel manifold.

Obviously the steepest descent algorithm requires the computation of the gradient. But the gradient is only defined after $T_X(n,p)$ is given an inner product:

$$< \boldsymbol{Z}_1, \boldsymbol{Z}_2 >= \Re\{tr(\boldsymbol{Z}_2^\dagger(\boldsymbol{I} - \frac{1}{2}\boldsymbol{X}\boldsymbol{X}^\dagger)\boldsymbol{Z}_1)\}, \tag{11}$$

where $\boldsymbol{Z}_1, \boldsymbol{Z}_2 \in T_X(n,p)$ and $\boldsymbol{X} \in St(n,p)$. The derivation of (11) can be found in [8]. Therefore, under the defined inner product, the steepest descent direction of the cost function $f(\boldsymbol{X})$ at the point $\boldsymbol{X} \in St(n,p)$ is

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{D}_X^\dagger\boldsymbol{X} - \boldsymbol{D}_X, \tag{12}$$

where $\boldsymbol{D}_X$ is the derivative of $f(\boldsymbol{X})$. The proof of (12) is in [9].

Algorithm 2 The Steepest Descent Algorithm on Complex Stiefel Manifold

Start with arbitrary precoder matrices $\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[K]}$, $\boldsymbol{V}^{[j]} \in St(M^{[j]},d^{[j]})$ and begin iteration.

for $j = 1...K$

(1) Compute the Jacobian matrix $\boldsymbol{D}_R^{[j]}$ and $\boldsymbol{D}_I^{[j]}$

(2) Then get the derivative of f:

$$\boldsymbol{D}_V^{[j]} = (\boldsymbol{D}_R^{[j]} + i\boldsymbol{D}_I^{[j]})^T$$

(3) Get the steepest descent direction

$$\boldsymbol{Z}^{[j]} = \boldsymbol{V}^{[j]}\boldsymbol{D}_V^{[j]\dagger}\boldsymbol{V}^{[j]} - \boldsymbol{D}_V^{[j]}$$

(4) Compute $\boldsymbol{B}_1^{[j]} = \pi(\boldsymbol{V}^{[j]} + 2\beta^{[j]}\boldsymbol{Z}^{[j]})$,
if $f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{K}) - f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[j-1]},\boldsymbol{B}_1^{[j]},\boldsymbol{V}^{[j+1]}...\boldsymbol{V}^{[K]}) \geq \beta^{[j]}\Re\{tr(\boldsymbol{Z}^{[j]\dagger}(\boldsymbol{I} - \frac{1}{2}\boldsymbol{V}^{[j]}\boldsymbol{V}^{[j]\dagger})\boldsymbol{Z}^{[j]})\}$, then set $\beta^{[j]} := 2\beta^{[j]}$, and repeat Step 4.

(5) Compute $\boldsymbol{B}_2^{[j]} = \pi(\boldsymbol{V}^{[j]} + \beta^{[j]}\boldsymbol{Z}^{[j]})$,
if $f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[K]}) - f(\boldsymbol{V}^{[1]},...,\boldsymbol{V}^{[j-1]},\boldsymbol{B}_2^{[j]},\boldsymbol{V}^{[j+1]}...\boldsymbol{V}^{[K]}) < \frac{1}{2}\beta^{[j]}\Re\{tr(\boldsymbol{Z}^{[j]\dagger}(I - \frac{1}{2}\boldsymbol{V}^{[j]}\boldsymbol{V}^{[j]\dagger})\boldsymbol{Z}^{[j]})\}$, then set $\beta^{[j]} := \frac{1}{2}\beta^{[j]}$, and repeat Step 5.

(6)$\boldsymbol{V}^{[j]} = \pi(\boldsymbol{V}^{[j]} + \beta^{[j]}\boldsymbol{Z}^{[j]})$

(7) Continue till the cost function f is sufficiently small.

Now the proposed SD algorithm on complex Stiefel manifold is presented in the table above. We also present some explanations here. From (11) and (12), it can be easily obtained that the inner product needed for the Armijo step rule is

$$\Re\{tr(\boldsymbol{Z}^{[j]\dagger}(\boldsymbol{I} - \frac{1}{2}\boldsymbol{V}^{[j]}\boldsymbol{V}^{[j]^\dagger})\boldsymbol{Z}^{[j]})\}, \tag{13}$$

which is used in Step 4 and 5; and the steepest descent on Stiefel manifold of our cost function is

$$\boldsymbol{Z}^{[j]} = \boldsymbol{V}^{[j]}\boldsymbol{D}_V^{[j]\dagger}\boldsymbol{V}^{[j]} - \boldsymbol{D}_V^{[j]}, \tag{14}$$

which is used in Step 3. Noticing that the project operation

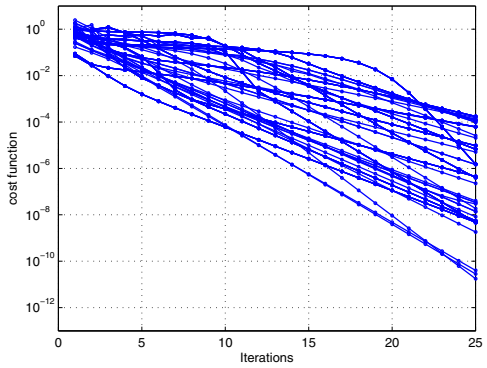

Fig. 2. Cost function of the SD algorithm in complex space

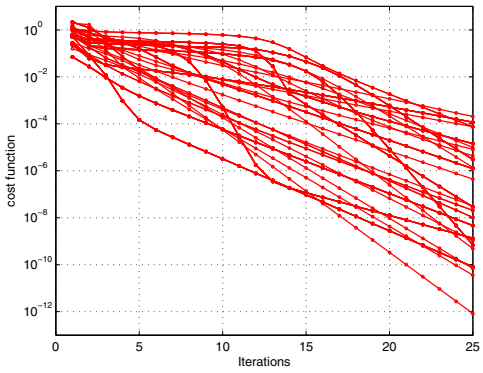

Fig. 3. Cost function of the SD algorithm on complex Stiefel manifold

$\pi(\cdot)$ in Step 6 (Step 4, 5) guarantees the newly computed solution $\boldsymbol{V}^{[j]}$ (or $\boldsymbol{B}_1^{[j]}$, $\boldsymbol{B}_2^{[j]}$) after iteration still satisfies $\boldsymbol{V}^{[j]} \in St(n,p)$. Using the method of SVD, we can easily compute the project operation.

IV. Numerical Results

Without symbol extension, the feasible condition of k-user inference alignment [3] is $M^{[j]} + N^{[j]} \geq (K+1)d^{[j]}$. Therefore, for satisfying feasibility and simple computation, we consider a 3-user 2x2 MIMO Interference Channel where the desired DoF per user $d^{[j]}$ is 1. We simulated the two SD algorithms through 100 randomly generated channel coefficients and initial precoder matrices. As shown in Fig. 2 and Fig. 3, each curve represents an individual simulation realization and all results converge after 20 or more iterations; the average value of the cost functions falls below 10^{-6}, which is a clear indication of the algorithm convergence performance in such limited iterations. Meanwhile, since there are two interference signals at each receiver, as shown in Fig. 4, the angles between the spaces spanned by each interference signals asymptotically converge to zero which is another strong evidence for achieving the perfect IA. Furthermore, in order to compare the convergence performance, within each realization of the simulation, all the algorithms are executed under the same scenario including randomly generated channel coefficients, initial precoder matrices and convergence step length. The mean values after 100 realizations are shown in Fig. 5. It can be observed that the SD algorithm on complex Stiefel manifold has better convergence performance. Finally we compare the system sum-rate of the proposed algorithms. Fig.6 shows that the SD algorithm in complex space and the Distributed IA in [4] almost have the same performance. And the SD algorithm on complex Stiefel manifold outperforms the other algorithms. More importantly, at high SNR, the DoF: $\frac{C_{sum}}{\log(SNR)}$ of the two proposed algorithms nearly achieve 3, which is the theoretical DoF ($\frac{KM}{2} = \frac{3\times 2}{2} = 3$) . Therefore the interference alignment is successfully achieved.

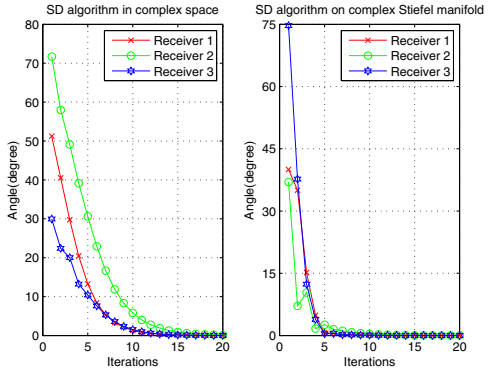

Fig. 4. Angle between interfering spaces at each receiver

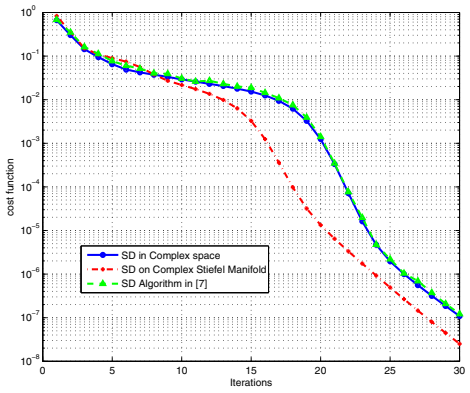

Fig. 5. Convergence performance

V. Discussion

As previous stated, comparing with the algorithm in [7], our modified SD algorithm in complex space has lower computing complexity and higher robustness. As shown in Fig. 5, the modified SD algorithm in complex space almost has the same convergence performance as algorithm in [7]. In addition, as shown in Fig. 6, the sum-rate curve of the SD algorithm in complex space and the curve of distributed IA in [4] almost overlap. While the authors in [7] claimed that, the capacity

performance of the work in [7] will not triumph over that of [4]. These two evidence guarantee that the modification does not degrade the algorithm performance.

More importantly, the novel SD algorithm on Stiefel manifold has better convergence performance. This is attributed to the reason that the proposed algorithm reformulates the constrained optimization problem to an unconstrained one on the complex Stiefel manifold and does not move descent step along the geodesic, as stated in the previous sections.

Two reasons leading to the fact that the SD algorithm on Stiefel manifold obtains higher system capacity (as shown in Fig. 6) are presented below:

- 1) It is noticed that our cost function actually is the interference power spilled from the interference space to the desired signal space. With the better convergence performance, the SD algorithm on Stiefel manifold will have less remanent interference in the desired signal space within same iteration times. Therefore, the SD algorithm on Stiefel manifold will get higher SINR [4]:

$$SINR = \frac{signal\ power}{noise + remanent\ interference} \tag{15}$$

  which leads to high capacity.
- 2) At each receiver, the zero forcing filter is adopted. It will project the desired signal power and the remanent interference onto the subspace which is orthogonal with the subspace spanned by the interference. After performing the SD algorithm on Stiefel manifold, it is observed that in the Euclidean norm distance, the subspace spanned by desired signal is more close to the orthogonal complement of the interference subspace (the result of this simulation is omitted due to the space limitation, and the reasons are deeply explored in journal version of this paper which is still in progress). Therefore, even the proposed two SD algorithms finally get the same remanent interference after several iterations. The SD algorithm on Stiefel manifold will lose less signal power during the projection operated by zero forcing filter, hence higher system capacity can be obtained.

The proof of convergence of the two proposed algorithms is intuitive. Our cost function is non-negative whose low bound is zero and monotonically decreases within each iteration. Therefore it must converge to a solution which is very close to zero. However, there is no guarantee that our cost function is convex [4]. Thus finding a global optimum is in our future work.

We also do some simulations by using more sophisticated algorithm, such as Newton-type method, to achieve quadratic convergence. Yet, except for the increased computational complexity, the Newton method will converge to the closet critical point (whether it is a local maximum, local minimum, or a saddle point) [9]. Therefore the Newton method coupled with the steepest descent algorithm (a few iterations of the SD algorithm are performed first to move close to a local minimum before the Newton algorithm is applied) will be investigated in our future work too.

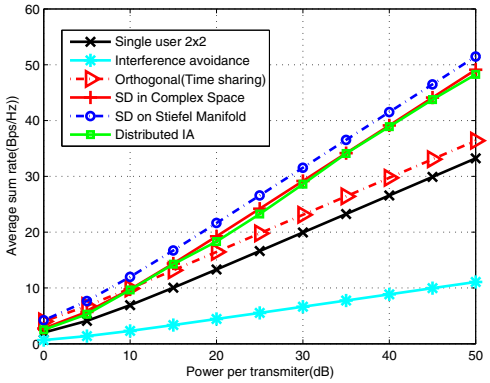

Fig. 6. Sum-rate capacity

VI. Conclusions

In this paper, we restrict the optimization only to the transmitters' side. A modified SD algorithm in complex space is proposed firstly. Then we reform the optimization problem on Stiefel manifold, and propose a novel SD algorithm on Stiefel manifold to achieve feasible interference alignment. Numerical simulation shows that comparing with previous methods, the novel SD algorithm on Stiefel manifold has better convergence performance and higher system capacity. Finally we prove that the algorithms converge monotonically with respect to the overall objective.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 61171112) and MIIT of China (No. 2010ZX03005-001-02).

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] M. Maddah-Ali, A. Motahari, and A. Khandani, "Signaling over MIMO multi-base systems: Combination of multi-access and broadcast schemes,"*Int. Symp. Inform Theory 2006*, pp. 2104-2108.

[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, pp. 5-8, Dec 2009.

[4] K. S. Gomadam, V. R. Cadambe, and S. A. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment,"*IEEE GLOBECOM*, New Orleans, Louisiana, Nov 2008, pp. 1-6.

[5] S. W. Peters, R. W. Heath, "Interference alignment via alternating minimization, " *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, April 2009, pp. 2445-2448.

[6] J. Raj Kumar and F. Xue, "An iterative algorithm for joint signal and interference alignment, "*IEEE ISIT 2010*, Austin, pp. 2293 - 2297

[7] Hadi G. Ghauch, Constantinos B. Papadias, "Interference alignment: A one-sided approach, "*Pro of IEEE GLOBECOM 2011*, Preprint available on arXiv.

[8] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthonormality constraints," *SIAM J. Matrix Anal. Applic.*, vol. 20, no. 2, pp.303 - 353 , 1998.

[9] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Trans. Signal Process*. 50 (2002), no. 3, 635-650.

[10] Xianda Zhang, *Matrix Analysis and applications*. Beijing, China: Springer, 2004.

[11] E. Polak, *Optimization: Algorithms and Consistent Approximations*. New York: Springer-Verlag, 1997.