

This Provisional PDF corresponds to the article as it appeared upon acceptance. Fully formatted PDF and full text (HTML) versions will be made available soon.

Low complexity interference alignment algorithms for desired signal power maximization problem of MIMO channels

EURASIP Journal on Advances in Signal Processing 2012,
2012:137 doi:10.1186/1687-6180-2012-137

Cong Sun (suncong@lsec.cc.ac.cn)
Yunchuan Yang (dachuan95@gmail.com)
Yaxiang Yuan (yyx@lsec.cc.ac.cn)

ISSN 1687-6180

Article type Research

Submission date 1 December 2011

Acceptance date 14 June 2012

Publication date 11 July 2012

Article URL <http://asp.eurasipjournals.com/content/2012/1/137>

This peer-reviewed article was published immediately upon acceptance. It can be downloaded, printed and distributed freely for any purposes (see copyright notice below).

For information about publishing your research in *EURASIP Journal on Advances in Signal Processing* go to

<http://asp.eurasipjournals.com/authors/instructions/>

For information about other SpringerOpen publications go to

<http://www.springeropen.com>

Low complexity interference alignment algorithms for desired signal power maximization problem of MIMO channels

Cong Sun^{1*}

*Corresponding author

Email: suncong@lsec.cc.ac.cn

Yunchuan Yang²

Email: dachuan95@gmail.com

Yaxiang Yuan¹

Email: yyx@lsec.cc.ac.cn

¹State Key Laboratory of Scientific and Engineering Computing, ICMSEC, AMSS, Chinese Academy of Sciences, Beijing, 100190, China

²Wireless Signal Processing and Network Lab, Beijing University of Posts and Telecommunications, Beijing, 100876, China

Abstract

In this article, we investigate the interference alignment (IA) solution for a K -user MIMO interference channel. Proper users' precoders and decoders are designed through a desired signal power maximization model with IA conditions as constraints, which forms a complex matrix optimization problem. We propose two low complexity algorithms, both of which apply the Courant penalty function technique to combine the leakage interference and the desired signal power together as the new objective function. The first proposed algorithm is the modified alternating minimization algorithm (MAMA), where each subproblem has closed-form solution with an eigenvalue decomposition. To further reduce algorithm complexity, we propose a hybrid algorithm which consists of two parts. As the first part, the algorithm iterates with Householder transformation to preserve the orthogonality of precoders and decoders. In each iteration, the matrix optimization problem is considered in a sequence of 2D subspaces, which leads to one dimensional optimization subproblems. From any initial point, this algorithm obtains precoders and decoders with low leakage interference in short time. In the second part, to exploit the advantage of MAMA, it continues to iterate to perfectly align the interference from the output point of the first part. Analysis shows that in one iteration generally both proposed two algorithms have lower computational complexity than the existed maximum signal power (MSP) algorithm, and the hybrid algorithm enjoys lower complexity than MAMA. Simulations reveal that both proposed algorithms achieve similar performances as the MSP algorithm with less executing time, and show better performances than the existed alternating minimization algorithm in terms of sum rate. Besides, from the view of convergence rate, simulation results show that the MAMA enjoys fastest speed with respect to a certain sum rate value, while hybrid algorithm converges fastest to eliminate interference.

Keywords

Interference alignment, Power maximization, Courant penalty function, Alternating minimization algorithm, Householder transformation

Introduction

Interference alignment (IA) technique is recently brought to show that each user can achieve half degree of freedom (DoF) in the K -user interference channel. It jointly optimizes precoding matrices for all transmitters, so that all interferences at one receiver fall into a reduced dimensional subspace. Then by multiplying decoding matrix orthogonal to this subspace, the certain receiver can extract the desired signals without interference. By utilizing the IA techniques, Cadambe and Jafar [1] showed that the achieved sum capacity of the K -user interference channel scales linearly with the number of users, in the high signal-to-noise-ratio (SNR) regime. Generally, the IA solutions are required to satisfy the following conditions simultaneously:

- (1) all the interferences are eliminated;
- (2) all the subspaces for desired signals are full rank;
- (3) precoders and decoders are required to be orthogonal.

The Yetis et al. [2] related the feasibility of the IA conditions in fully connected interference channel to the problem of determining the solvability of a multivariate polynomial system, with arbitrary antenna configurations. The achievable DoFs are also discussed based on this polynomial system, relating to users' number of antennas and the number of users [3]. The analysis is further extended to that in partially connected channels [4–6].

From the signal processing point of view, the procedure of IA is to solve precoders and decoders according to the three conditions with a feasible IA system. However, the solution to this feasible problem is still not known in general. There are available closed form solutions only for certain cases, such as 3-user MIMO channel with N antennas each user equips with and $N/2$ DoFs each user requires, and K -user channel where each user equips with $K - 1$ antennas and wishes to achieve 1 DoF. For general cases, the system is turned into an optimization problem minimizing the total leakage interference and preserving the orthogonality of precoders and decoders as constraints, which is denoted as leakage interference minimization (LIM) problem. With the solution, the IA condition 2 can be almost surely satisfied if channels have no special structures [7]. LIM problem is proved to be NP-hard when the number of antennas each user equips with are greater than 3 [8]. Thus, iterative algorithms rather than the analytical solutions should be considered. Gomadam et al. [7] exploited the channel reciprocity and proposed an alternating minimization algorithm (AMA) to design precoders and decoders in a distributed way. Then each subproblem is equivalent to an eigenvalue problem with eigenvalue decomposition required. In the AMA, although the leakage interference can be perfectly canceled after convergence, its performance in terms of sum rate is not optimal. In fact, it is pointed out that for general interference channels the constructed LIM problem has a large number of different IA solutions obtained from different initial points, which lead to different achieved sum rate values [9]. The main reason is that the AMA only eliminates the interference in the desired signal space without considering the system sum rate, which results in a suboptimal sum rate achieved with finite signal power.

Gomadam et al. [7] also noticed the disadvantage of AMA, and then proposed the Max signal-to-interference-and-noise-ratio (SINR) algorithm as well. In each iteration, the basic idea of the Max-SINR algorithm is to choose the precoders and decoders stream by stream, with aim to maximize the SINR of each stream instead of minimizing the leakage interference. Due to the relaxation of the IA condition 3, the IA condition 2 does not hold anymore in this algorithm, which means the required DoFs might not be satisfied. This analysis accords with the performances shown in [7], that it achieves higher sum rate than the AMA in intermediate SNR regime, however suffers from the loss of required DoFs in the high SNR regime. Many other algorithms can also be designed to perform like Max-SINR, however none of them can achieve the optimal DoFs without IA conditions [10]. Further, IA scheme provides the receivers interference-free subspaces, with which receivers completely get rid of complicated cancelation of interference. Thus, it is important to improve IA algorithms in general SNR scenarios.

Prior study in [11] proposes an iterative algorithm using the gradient descent method to solve the new IA model, where the utility function of either sum rate or the desired signal power is maximized with the IA conditions 1 and 3 as constraints. The corresponding MSP algorithm is shown to obtain higher sum rate than the AMA generally regardless of the initial point and higher than Max-SINR in high SNR regime. However, it requires a series of eigenvalue decompositions and compact singular value decompositions (SVDs), which lead to high computational complexity. Besides, the MSP algorithm has much slower convergence rate than the AMA.

Nevertheless, in practical systems, receivers have limited computational complexity, which might be a bottleneck for the complexity of the algorithms. Besides, channel reciprocity requires TDD operations which restrict the executing time of the algorithms. Therefore, the principal question here is how to design proper algorithms to balance the achieved sum rate with computational complexity and computing time. In this article, we aim to propose algorithms to maintain the advantage of the MSP algorithm with faster convergence, lower complexity and less executing time. Two efficient algorithms are proposed to solve the desired signal power maximization IA problem. First we propose a modified alternating minimization algorithm (MAMA) with Courant penalty function technique. Then, to further reduce the algorithm complexity, a new algorithm with Householder transformation (AHT) is proposed, where a two-dimensional subspace method is applied to solve subproblems. It acts to obtain precoders and decoders with low leakage interference in short time, and then the MAMA continues iterating to get perfect IA, which forms the hybrid algorithm.

The remaining article is organized as follows: In Section “Desired signal power maximization interference alignment model”, the desired signal power maximization model is presented. The two algorithms, MAMA and hybrid algorithm, are proposed in Sections “Modified alternating minimization algorithm” and “A hybrid algorithm”, respectively. The computational complexity of the proposed algorithms and MSP algorithm are analyzed in Section “Analysis of computational complexity”. Numerical results and further remarks are shown in Section “Simulations”.

Notation: Lowercase and uppercase boldface represent vectors and matrices, respectively. \mathbb{C} represents the complex domain. $\text{Re}(a)$ means the real part of scalar a . $\text{tr}(\mathbf{A})$ and $\|\mathbf{A}\|_F$ are the trace and the Frobenius norm of matrix \mathbf{A} , respectively. \mathbf{I}_d represents the $d \times d$ identity matrix. \mathcal{K} represents the set of the user indices $\{1, 2, \dots, K\}$. $\mathcal{CN}(\mu, \sigma^2)$ means the complex Gaussian distribution with mean μ and variance σ^2 . And we use $E_{\mathbf{X}}(\cdot)$ to denote the statistical expectation with the variable \mathbf{X} . $O(n)$ means the same order amount of n .

Desired signal power maximization IA model

Consider a K -user interference MIMO channel $(M \times N, d)^K$ as in Figure 1, where each transmitter equips with M antennas, each receiver with N antennas and each user pair wishes to achieve d DoFs. Suppose $\mathbf{s}_k \in \mathbb{C}^{d \times 1}$ denotes the transmit signal vector of the k th user with power covariance as $E(\mathbf{s}_k \mathbf{s}_k^H) = (P/d)\mathbf{I}_d$, where P is the total transmit power of each user. For convenience we unify the transmit power of each stream, i.e. $P/d = 1$. After receiving the signal \mathbf{y}_k , the k th receiver multiplies decoding matrix to it on the left, which is expressed as:

$$\mathbf{U}_k^H \mathbf{y}_k = \underbrace{\mathbf{U}_k^H \mathbf{H}_{kk} \mathbf{V}_k \mathbf{s}_k}_{\text{desired signal}} + \underbrace{\sum_{l \neq k, l \in \mathcal{K}} \mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l \mathbf{s}_l}_{\text{interference}} + \underbrace{\mathbf{U}_k^H \mathbf{n}_k}_{\text{noise}}, \quad (1)$$

where $\mathbf{H}_{kl} \in \mathbb{C}^{N \times M}$ denotes the channel matrix between the k th transmitter and the l th receiver, and $\mathbf{V}_k \in \mathbb{C}^{M \times d}$, $\mathbf{U}_k \in \mathbb{C}^{N \times d}$ represent the precoder and decoder of the k th user, respectively. The three terms of (1) on the right side represent the desired signal, the interference from other users and the noise with distribution of $\mathcal{CN}(0, \sigma_k^2 \mathbf{I}_N)$ at the k th receiver, respectively. In this article we assume all the noises have the same covariance, that is $\sigma_k^2 = \sigma^2, k \in \mathcal{K}$.

Figure 1 K -user MIMO interference channel

Following the feasible condition for IA system, IA scheme is defined as [7]:

$$\mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l = \mathbf{0}, k \neq l, k, l \in \mathcal{K}; \quad (2)$$

$$\text{rank}(\mathbf{U}_k^H \mathbf{H}_{kk} \mathbf{V}_k) = d, k \in \mathcal{K}; \quad (3)$$

$$\mathbf{V}_k^H \mathbf{V}_k = \mathbf{I}_d, \mathbf{U}_k^H \mathbf{U}_k = \mathbf{I}_d, k \in \mathcal{K}. \quad (4)$$

The original idea of IA only consists of (2) and (3). Noticing that for any precoding and decoding matrices $\{\mathbf{V}_k, \mathbf{U}_k, k \in \mathcal{K}\}$ that satisfy these two conditions, $\{\mathbf{V}_k \mathbf{P}_k, \mathbf{U}_k \mathbf{Q}_k, k \in \mathcal{K}\}$ also satisfy (2) and (3) as long as $\{\mathbf{P}_k, \mathbf{Q}_k \in \mathbb{C}^{d \times d}, k \in \mathcal{K}\}$ are all non-singular matrices. This indicates that the solutions of the IA system are not unique and the solution matrices form d -dimensional subspaces. Therefore, we require the columns of $\mathbf{V}_k, \mathbf{U}_k, k \in \mathcal{K}$ to be the orthogonal basis of the corresponding subspaces, which is the condition (4).

Besides the requirement of (2) and (4), we wish to maximize the desired signal power, in order to achieve sufficiently high sum rate. Suppose all the transmit signals are statistically independent of each other. We can obtain the expected total desired signal power

$$P^S = \sum_{k \in \mathcal{K}} E_s(\|\mathbf{U}_k^H \mathbf{H}_{kk} \mathbf{V}_k \mathbf{s}_k\|_2^2) = \sum_{k \in \mathcal{K}} \|\mathbf{U}_k^H \mathbf{H}_{kk} \mathbf{V}_k\|_F^2.$$

Based on the above analysis, we present the desired signal power maximization (PM) model as follows:

$$\max_{\substack{\mathbf{U}_k, \mathbf{V}_k \\ k \in \mathcal{K}}} P^S(\mathbf{U}_k, \mathbf{V}_k) = \sum_{k \in \mathcal{K}} \|\mathbf{U}_k^H \mathbf{H}_{kk} \mathbf{V}_k\|_F^2 \quad (5a)$$

$$\text{s. t. } \mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l = \mathbf{0}, \quad k \neq l, k, l \in \mathcal{K}, \quad (5b)$$

$$\mathbf{V}_k^H \mathbf{V}_k = \mathbf{I}_d, \mathbf{U}_k^H \mathbf{U}_k = \mathbf{I}_d, \quad k \in \mathcal{K}. \quad (5c)$$

This model was first brought in [11], in which its performance is compared with the sum rate maximization (SRM) model. The PM model can achieve similar performance as the SRM model, while its related optimization problem is much simpler. Therefore it is a good way to approximate the SRM model by the PM model [11]. Thus in this article we only focus on solving (5) to design proper precoders and decoders.

The sum of squares of the residuals of constraints (5b) is the total leakage interference, which is given by:

$$P^l(\mathbf{U}_k, \mathbf{V}_k) = \sum_{k \in \mathcal{K}} E_s \left(\left\| \sum_{l \neq k, l \in \mathcal{K}} \mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l \mathbf{s}_l \right\|^2 \right) = \sum_{k \in \mathcal{K}} \sum_{l \neq k, l \in \mathcal{K}} \|\mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l\|_F^2. \quad (6)$$

The essential idea of the AMA given by [7] is to minimize P^l alternately for \mathbf{U}_k and \mathbf{V}_k . From the formulation of (5), we know that the MSP algorithm in [11] solves it by increasing P^S while reducing P^l towards zero at the same time. Here we briefly introduce the framework of MSP. In each iteration for each precoder and decoder, an AMA step is first taken to reduce P^l . And then the iterative point go along the gradient direction of P^S to increase it, whose stepsize gradually shrinks to 0 in order to perfectly align the interference by simply taking AMA steps in the last few iterations. MSP algorithm requires a series of eigenvalue decompositions and SVDs, which bring in high computational complexity and complicated matrix computations. Therefore we wish to propose algorithms for (5) with lower complexity and less complicated computations.

Modified alternating minimization algorithm (MAMA)

In this section, we present our first low complexity algorithm, the MAMA. The main difficulty for solving the highly nonlinear nonconvex optimization problem (5) is to deal with the nonlinear constraint (5b). Noticing that (5a) and the penalty term (6) for (5b) have quite similar expressions, we can combine these two together by the Courant penalty function technique [12] as in the next section.

Courant penalty function technique

Courant penalty function technique is a classic penalty function technique, which avoids dealing with constraints by moving them to the objective function. The basic idea is to replace the constraints with a penalty term scaled by a parameter C . The penalty term is the sum of squares of all the constraint violations.

In (5), we apply Courant penalty technique to (5b) and keep the constraints (5c) in order to obtain the following simple constrained problem:

$$\min_{\substack{\mathbf{U}_k, \mathbf{V}_k \\ k \in \mathcal{K}}} P(\mathbf{U}_k, \mathbf{V}_k, C) = CP^l(\mathbf{U}_k, \mathbf{V}_k) - P^S(\mathbf{U}_k, \mathbf{V}_k) \quad (7a)$$

$$\text{s. t. } \mathbf{V}_k^H \mathbf{V}_k = \mathbf{I}_d, \mathbf{U}_k^H \mathbf{U}_k = \mathbf{I}_d, \quad k \in \mathcal{K}, \quad (7b)$$

where P^l is defined by (6).

It is well known that if (5) is feasible, the solution of (7) converges to that of (5) as C approaches infinity [13].

Thus the solution of (5) can be approximated by that of (7). As pointed out in [13], the approximated error between the solutions of (7) and (5) is of the level $O(\frac{1}{C})$, which concerns about the choice of the penalty parameter C . Therefore, theoretically, (7) with larger C would get better approximation of (5). However, in real computations, large C may lead to ill-conditioned objective function and consequently bring in numerical calculation difficulties. To avoid such difficulties caused by unnecessarily large C in the first few iterations, we initially set C as a small positive number such as $C = 1$. If sufficient reduction in P^l is not achieved, we increase C to $\sigma_0 C$, where $\sigma_0 > 1$ is a constant. The solution of (7), $(\mathbf{U}_k(C), \mathbf{V}_k(C), k \in \mathcal{K})$ provides a good initial point for solving (7) again when C is replaced by $\sigma_0 C$. In this way we force P^l to reduce towards 0 eventually and the optimal solution of (5) can be obtained.

The overall algorithm

Via Courant penalty function technique our main task is now how to solve (7), which is quite similar to the LIM problem in which P^l is to be minimized. We can rewrite (7) as:

$$\begin{aligned} \min_{\substack{\mathbf{U}_k, \mathbf{V}_k \\ k \in \mathcal{K}}} P(\mathbf{U}_k, \mathbf{V}_k, C) & \quad (8a) \\ &= \sum_{k \in \mathcal{K}} \text{tr} \left(\mathbf{U}_k^H \left(C \sum_{l \neq k, l \in \mathcal{K}} \mathbf{H}_{kl} \mathbf{V}_l \mathbf{V}_l^H \mathbf{H}_{kl}^H - \mathbf{H}_{kk} \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_{kk}^H \right) \mathbf{U}_k \right) \\ &= \sum_{k \in \mathcal{K}} \text{tr} \left(\mathbf{V}_k^H \left(C \sum_{l \neq k, l \in \mathcal{K}} \mathbf{H}_{lk}^H \mathbf{U}_l \mathbf{U}_l^H \mathbf{H}_{lk} - \mathbf{H}_{kk}^H \mathbf{U}_k \mathbf{U}_k^H \mathbf{H}_{kk} \right) \mathbf{V}_k \right) \\ \text{s. t. } \mathbf{V}_k^H \mathbf{V}_k &= \mathbf{I}_d, \mathbf{U}_k^H \mathbf{U}_k = \mathbf{I}_d, \quad k \in \mathcal{K}. & \quad (8b) \end{aligned}$$

The only difference between the LIM problem and (8) is the objective function, as LIM minimizes the total leakage interference P^l subject to (8b). Thus, we can borrow the idea of the AMA in [7], to iterate $\mathbf{V}_k, k \in \mathcal{K}$ and $\mathbf{U}_k, k \in \mathcal{K}$ alternatively in each iteration. Fixing all $\mathbf{V}_k, k \in \mathcal{K}$ in (8), $\mathbf{U}_k, k \in \mathcal{K}$ become independent of each other. In this case, (8) turns into K independent subproblems with formulas as:

$$\begin{aligned} \min_{\mathbf{U}_k} \text{tr} \left(\mathbf{U}_k^H \left(C \sum_{l \neq k, l \in \mathcal{K}} \mathbf{H}_{kl} \mathbf{V}_l \mathbf{V}_l^H \mathbf{H}_{kl}^H - \mathbf{H}_{kk} \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_{kk}^H \right) \mathbf{U}_k \right) & \\ \text{s. t. } \mathbf{U}_k^H \mathbf{U}_k &= \mathbf{I}_d. & \quad (9) \end{aligned}$$

We can obtain the closed-form solution of (9) as $\mathbf{U}_k = v_{\min}^d(\mathbf{J}_k), k \in \mathcal{K}$, which means the columns of \mathbf{U}_k are eigenvectors corresponding to the d smallest eigenvalues of

$$\mathbf{J}_k = C \sum_{l \neq k} \mathbf{H}_{kl} \mathbf{V}_l \mathbf{V}_l^H \mathbf{H}_{kl}^H - \mathbf{H}_{kk} \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_{kk}^H.$$

Similar solutions of \mathbf{V}_k can be achieved when fixing all $\mathbf{U}_k, k \in \mathcal{K}$. Of course, we need a technique to update the penalty parameter C according to P^l during iterations.

Based on the above descriptions, our MAMA for the PM model (5) is stated as follows (Algorithm 1).

Algorithm 1: modified alternating minimization algorithm

1. Set initial precoders \mathbf{V}_k and decoders $\mathbf{U}_k, k \in \mathcal{K}$, the initial penalty parameter C and $\sigma_0 > 1$.
 $\tilde{P}^l = P^l(\mathbf{U}_k, \mathbf{V}_k)$.
2. For decoders: $\mathbf{U}_k = v_{\min}^d(\mathbf{J}_k), k \in \mathcal{K}$ from above.
3. For precoders: $\mathbf{V}_k = v_{\min}^d(\tilde{\mathbf{J}}_k), k \in \mathcal{K}$, where $\tilde{\mathbf{J}}_k = C \sum_{l \neq k} \mathbf{H}_{lk}^H \mathbf{U}_l \mathbf{U}_l^H \mathbf{H}_{lk} - \mathbf{H}_{kk}^H \mathbf{U}_k \mathbf{U}_k^H \mathbf{H}_{kk}$.
4. If the algorithm converges, then stop and output $\mathbf{V}_k, \mathbf{U}_k, k \in \mathcal{K}$. If $P^l(\mathbf{U}_k, \mathbf{V}_k) > \tilde{P}^l - 1/C$, then increase the penalty parameter $C := \sigma_0 C$. If $P^l(\mathbf{U}_k, \mathbf{V}_k) < \tilde{P}^l$, let $\tilde{P}^l = P^l(\mathbf{U}_k, \mathbf{V}_k)$. Go to step 2.

In the original AMA of [7], eigenvectors corresponding to the d smallest eigenvalues of positive semi-definite matrices $\{\sum_{l \neq k} \mathbf{H}_{kl} \mathbf{V}_l \mathbf{V}_l^H \mathbf{H}_{kl}^H, k \in \mathcal{K}\}$ and those of $\{\sum_{l \neq k} \mathbf{H}_{kl}^H \mathbf{U}_l \mathbf{U}_l^H \mathbf{H}_{kl}, k \in \mathcal{K}\}$ are required. In our MAMA, the matrices \mathbf{J}_k and $\tilde{\mathbf{J}}_k$ are not necessarily positive semi-definite, since the objective function value of (8) might be negative with feasible solution of quite low P^I and high P^S .

Although our MAMA avoids calculating SVD which is required by the MSP algorithm given by [11], eigenvalue decompositions are still required for solving subproblems. This lends an impetus to further improvement.

A hybrid algorithm

In order to further reduce the complexity of our MAMA, we propose a hybrid technique in this section. First, we give an AHT, which is a very low complexity algorithm for (5). This algorithm is free from complicated matrix computation, such as eigenvalue decomposition, SVD and QR factorization, and enjoys lower computational complexity than MAMA.

Algorithm with Householder transformation

In this algorithm, we still use the Courant penalty function technique. Thus we focus on problem (7). The only constraints of (7) are orthogonal constraints. We require all the iterative points \mathbf{U}_k and \mathbf{V}_k generated by our algorithms are feasible, namely (7b) are always satisfied. By requiring feasibility, basically we are solving an unconstrained optimization problem on the Grassmann manifold. Otherwise, if we allow iterations go outside the feasible region, we will have to adopt some technique, such as projection, to draw the iterations back to the feasible region, which can be complicated. For example, the projection to the set of orthogonal matrices can be a non-differentiable operator and time-consuming to compute.

Preserving orthogonality

Let \mathbf{V}_k^i denotes the precoder of the k th user in the i th iteration. Suppose both \mathbf{V}_k^i and \mathbf{V}_k^{i+1} are orthogonal, there should exist a unitary matrix $\mathbf{P}_k^i \in \mathbb{C}^{M \times M}$ such that

$$\mathbf{P}_k^i \mathbf{V}_k^i = \mathbf{V}_k^{i+1}.$$

The straightforward approach is to search for all unitary matrices $\mathbf{P}_k^i \in \mathbb{C}^{M \times M}$, but this can be computationally very expensive. We consider a special class of unitary matrices, the Householder matrices, which can be represented as $\mathbf{P} = \mathbf{I} - 2\mathbf{p}\mathbf{p}^H$ with $\mathbf{p}^H \mathbf{p} = 1$. The use of Householder transformation can not only preserve orthogonality, but also turn the problem from a matrix optimization into a vector optimization problem.

In the i th iteration with fixed $\mathbf{V}_k^i, \mathbf{U}_k^i, k \in \mathcal{K}$ as certain feasible solution of (7), we try to obtain orthogonal vectors $\mathbf{v}_k, \mathbf{u}_k, k \in \mathcal{K}$ and set

$$\mathbf{V}_k^{i+1} = (\mathbf{I}_M - 2\mathbf{v}_k \mathbf{v}_k^H) \mathbf{V}_k^i, \mathbf{U}_k^{i+1} = (\mathbf{I}_N - 2\mathbf{u}_k \mathbf{u}_k^H) \mathbf{U}_k^i, \quad k \in \mathcal{K} \quad (10)$$

as the precoders and decoders in the $(i+1)$ th iteration. Here, $\mathbf{v}_k, \mathbf{u}_k, k \in \mathcal{K}$ are regarded as optimal (or suboptimal) solutions to minimize the objective function value of (7),

$$P(\mathbf{U}_k^{i+1}, \mathbf{V}_k^{i+1}, C) = P((\mathbf{I}_N - 2\mathbf{u}_k \mathbf{u}_k^H) \mathbf{U}_k^i, (\mathbf{I}_M - 2\mathbf{v}_k \mathbf{v}_k^H) \mathbf{V}_k^i, C).$$

As precoding and decoding matrices are updated according to Householder transformation in each iteration, the new algorithm is denoted as AHT.

Alternating directions

Substituting (10) into (7), we get (11) with variables $\mathbf{v}_k, \mathbf{u}_k, k \in \mathcal{K}$:

$$\begin{aligned}
& \min_{\substack{\mathbf{u}_k, \mathbf{v}_k \\ k \in \mathcal{K}}} \sum_{k \in \mathcal{K}} \left(C \sum_{l \neq k, l \in \mathcal{K}} \|(\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{V}_k^{i+1}\|_F^2 - \|(\mathbf{U}_k^{i+1})^H \mathbf{H}_{kk} \mathbf{V}_k^{i+1}\|_F^2 \right) \\
& = \sum_{k \in \mathcal{K}} \left(C \sum_{l \neq k} \|(\mathbf{U}_l^i)^H (\mathbf{I}_N - 2\mathbf{u}_l \mathbf{u}_l^H) \mathbf{H}_{lk} (\mathbf{I}_M - 2\mathbf{v}_k \mathbf{v}_k^H) \mathbf{V}_k^i\|_F^2 \right. \\
& \quad \left. - \|(\mathbf{U}_k^i)^H (\mathbf{I}_N - 2\mathbf{u}_k \mathbf{u}_k^H) \mathbf{H}_{kk} (\mathbf{I}_M - 2\mathbf{v}_k \mathbf{v}_k^H) \mathbf{V}_k^i\|_F^2 \right) \\
& \text{s. t. } \mathbf{v}_k^H \mathbf{v}_k = 1, \mathbf{u}_k^H \mathbf{u}_k = 1, \quad k \in \mathcal{K}.
\end{aligned} \tag{11}$$

With the fact that

$$\begin{aligned}
& \|(\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} (\mathbf{I}_M - 2\mathbf{v}_k \mathbf{v}_k^H) \mathbf{V}_k^i\|_F^2 \\
& = \text{tr}((\mathbf{V}_k^i)^H (\mathbf{I}_M - 2\mathbf{v}_k \mathbf{v}_k^H) \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} (\mathbf{I}_M - 2\mathbf{v}_k \mathbf{v}_k^H) \mathbf{V}_k^i) \\
& = 4\text{tr}((\mathbf{V}_k^i)^H \mathbf{v}_k \mathbf{v}_k^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{v}_k \mathbf{v}_k^H \mathbf{V}_k^i) - 2\text{tr}((\mathbf{V}_k^i)^H \mathbf{v}_k \mathbf{v}_k^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{V}_k^i \\
& \quad + (\mathbf{V}_k^i)^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{v}_k \mathbf{v}_k^H \mathbf{V}_k^i) + \text{tr}((\mathbf{V}_k^i)^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{V}_k^i) \\
& = 4(\mathbf{v}_k^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{v}_k)(\mathbf{v}_k^H \mathbf{V}_k^i (\mathbf{V}_k^i)^H \mathbf{v}_k) - 2(\mathbf{v}_k^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{V}_k^i (\mathbf{V}_k^i)^H \mathbf{v}_k \\
& \quad + \mathbf{v}_k^H \mathbf{V}_k^i (\mathbf{V}_k^i)^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{v}_k) + \text{tr}((\mathbf{V}_k^i)^H \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} \mathbf{V}_k^i),
\end{aligned} \tag{12}$$

we can rewrite the objective function of (11) as:

$$\sum_{k \in \mathcal{K}} [4(\mathbf{v}_k^H \mathbf{A}_k \mathbf{v}_k)(\mathbf{v}_k^H \mathbf{V}_k^i (\mathbf{V}_k^i)^H \mathbf{v}_k) - 2(\mathbf{v}_k^H \mathbf{A}_k \mathbf{V}_k^i (\mathbf{V}_k^i)^H \mathbf{v}_k + \mathbf{v}_k^H \mathbf{V}_k^i (\mathbf{V}_k^i)^H \mathbf{A}_k \mathbf{v}_k) + \text{tr}((\mathbf{V}_k^i)^H \mathbf{A}_k \mathbf{V}_k^i)], \tag{13}$$

where $\mathbf{A}_k = C \sum_{l \neq k, l \in \mathcal{K}} \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} - \mathbf{H}_{kk}^H \mathbf{U}_k^{i+1} (\mathbf{U}_k^{i+1})^H \mathbf{H}_{kk}, k \in \mathcal{K}$.

Fixing all $\mathbf{u}_l, l \in \mathcal{K}$ in (11), $\mathbf{U}_l^{i+1}, l \in \mathcal{K}$ are determined according to (10), and thus $\mathbf{v}_k, k \in \mathcal{K}$ become independent of each other. Therefore, (11) is decomposed into K subproblems with the same form as follows:

$$\begin{aligned}
& \min_{\mathbf{x} \in \mathbb{C}^M} f(\mathbf{x}) = (\mathbf{x}^H \mathbf{A} \mathbf{x})(\mathbf{x}^H \mathbf{B} \mathbf{x}) - \mathbf{x}^H \mathbf{C} \mathbf{x} \\
& \text{s. t. } \mathbf{x}^H \mathbf{x} = 1.
\end{aligned} \tag{14}$$

Here \mathbf{x} represents \mathbf{v}_k . $\mathbf{A} = 2(C \sum_{l \neq k, l \in \mathcal{K}} \mathbf{H}_{lk}^H \mathbf{U}_l^{i+1} (\mathbf{U}_l^{i+1})^H \mathbf{H}_{lk} - \mathbf{H}_{kk}^H \mathbf{U}_k^{i+1} (\mathbf{U}_k^{i+1})^H \mathbf{H}_{kk}), \mathbf{B} = 2\mathbf{V}_k^i (\mathbf{V}_k^i)^H$ and $\mathbf{C} = (\mathbf{A}\mathbf{B} + \mathbf{B}\mathbf{A})/2$. Similarly, fixing all $\mathbf{v}_k, k \in \mathcal{K}$, it yields K subproblems like (14) from (11). Based on the analysis above, the alternating direction method can be applied to optimize $\mathbf{v}_k, k \in \mathcal{K}$ and $\mathbf{u}_k, k \in \mathcal{K}$ alternatively.

Subproblem: 2D subspace method

Subproblem (14) is a nonlinear optimization on the Stiefel manifold in complex field, which is not easy to handle. With the aim of low complexity, we abandon the second order method where Hessian matrix information is required. Instead, we apply a 2D subspace method to solve it iteratively [14]. We define the subspace spanned by \mathbf{x}_0 and $\tilde{\mathbf{g}}$ as

$$S(\mathbf{x}_0, \tilde{\mathbf{g}}) = \left\{ \mathbf{x} : \mathbf{x} = b\mathbf{x}_0 + a\tilde{\mathbf{g}}, a \in [-1, 1], b = \sqrt{1 - a^2} \right\},$$

where \mathbf{x}_0 is the current iterative feasible point and $\tilde{\mathbf{g}}$ represents its normalized gradient of the Lagrangian function of (14). In each step, the feasible domain of \mathbf{x} shrinks from the Stiefel manifold $\mathbf{x}^H \mathbf{x} = 1$ to $S(\mathbf{x}_0, \tilde{\mathbf{g}})$.

Here $\tilde{\mathbf{g}}$ is calculated as follows. First, the Lagrangian function of (14) is expressed as:

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda(\mathbf{x}^H \mathbf{x} - 1), \quad (15)$$

where λ is the Lagrange multiplier. Then the gradient of (15) is

$$\mathbf{g} = \mathbf{g}_0 - \lambda \mathbf{x}_0, \quad (16)$$

where \mathbf{g}_0 is the gradient of the objective function f at the point \mathbf{x}_0 :

$$\mathbf{g}_0 = 2[(\mathbf{x}_0^H \mathbf{A} \mathbf{x}_0) \mathbf{B} \mathbf{x}_0 + (\mathbf{x}_0^H \mathbf{B} \mathbf{x}_0) \mathbf{A} \mathbf{x}_0 - \mathbf{C} \mathbf{x}_0].$$

As a necessary condition of the first order optimality condition of (14), we require $\mathbf{g}^H \mathbf{x}_0 = 0$. Thus $\lambda = 2(\mathbf{x}_0^H \mathbf{A} \mathbf{x}_0)(\mathbf{x}_0^H \mathbf{B} \mathbf{x}_0) - \mathbf{x}_0^H \mathbf{C} \mathbf{x}_0$ is deduced due to $\mathbf{x}_0^H \mathbf{x}_0 = 1$. Once λ is computed, \mathbf{g} can be chosen by (16) and we can let $\tilde{\mathbf{g}} = \mathbf{g}/\|\mathbf{g}\|_2$.

To maintain $S(\mathbf{x}_0, \tilde{\mathbf{g}})$ on the Stiefel manifold, any point $\mathbf{x} \in S(\mathbf{x}_0, \tilde{\mathbf{g}})$ should satisfy $\mathbf{x}^H \mathbf{x} = b^2 \|\mathbf{x}_0\|^2 + a^2 \|\tilde{\mathbf{g}}\|^2 = 1$. Thus $a^2 + b^2 = 1$, $a \in [-1, 1]$, namely $b = \pm\sqrt{1 - a^2}$. For any $a \in [-1, 1]$, the objective function of (14) share the same function value at $\sqrt{1 - a^2} \mathbf{x}_0 + a \tilde{\mathbf{g}}$ and $-\sqrt{1 - a^2} \mathbf{x}_0 - a \tilde{\mathbf{g}}$. Therefore, we only consider the case $b = \sqrt{1 - a^2}$ to avoid redundant solutions.

Taking the expression of $\mathbf{x} \in S(\mathbf{x}_0, \tilde{\mathbf{g}})$ into (14), it turns into a one dimensional constraint optimization problem with variable a :

$$\min_{a \in [-1, 1]} p_1 a \sqrt{1 - a^2} + p_2 a^2 + p_3 a^3 \sqrt{1 - a^2} + p_4 a^4, \quad (17)$$

where the current iterative point \mathbf{x}_0 corresponds to $a = 0$, and

$$\begin{aligned} p_1 &= 2[\operatorname{Re}(\mathbf{x}_0^H \mathbf{A} \tilde{\mathbf{g}})(\mathbf{x}_0^H \mathbf{B} \mathbf{x}_0) + \operatorname{Re}(\mathbf{x}_0^H \mathbf{B} \tilde{\mathbf{g}})(\mathbf{x}_0^H \mathbf{A} \mathbf{x}_0) - \operatorname{Re}(\mathbf{x}_0^H \mathbf{C} \tilde{\mathbf{g}})], \\ p_2 &= (\mathbf{x}_0^H \mathbf{A} \mathbf{x}_0)(\tilde{\mathbf{g}}^H \mathbf{B} \tilde{\mathbf{g}}) + (\mathbf{x}_0^H \mathbf{B} \mathbf{x}_0)(\tilde{\mathbf{g}}^H \mathbf{A} \tilde{\mathbf{g}}) - 2(\mathbf{x}_0^H \mathbf{A} \mathbf{x}_0)(\mathbf{x}_0^H \mathbf{B} \mathbf{x}_0) \\ &\quad + 4\operatorname{Re}(\mathbf{x}_0^H \mathbf{A} \tilde{\mathbf{g}})\operatorname{Re}(\mathbf{x}_0^H \mathbf{B} \tilde{\mathbf{g}}) + \mathbf{x}_0^H \mathbf{C} \mathbf{x}_0 - \tilde{\mathbf{g}}^H \mathbf{C} \tilde{\mathbf{g}}, \\ p_3 &= 2[\operatorname{Re}(\mathbf{x}_0^H \mathbf{B} \tilde{\mathbf{g}})(\tilde{\mathbf{g}}^H \mathbf{A} \tilde{\mathbf{g}} - \mathbf{x}_0^H \mathbf{A} \mathbf{x}_0) + \operatorname{Re}(\mathbf{x}_0^H \mathbf{A} \tilde{\mathbf{g}})(\tilde{\mathbf{g}}^H \mathbf{B} \tilde{\mathbf{g}} - \mathbf{x}_0^H \mathbf{B} \mathbf{x}_0)], \\ p_4 &= (\tilde{\mathbf{g}}^H \mathbf{A} \tilde{\mathbf{g}} - \mathbf{x}_0^H \mathbf{A} \mathbf{x}_0)(\tilde{\mathbf{g}}^H \mathbf{B} \tilde{\mathbf{g}} - \mathbf{x}_0^H \mathbf{B} \mathbf{x}_0) - 4\operatorname{Re}(\mathbf{x}_0^H \mathbf{A} \tilde{\mathbf{g}})\operatorname{Re}(\mathbf{x}_0^H \mathbf{B} \tilde{\mathbf{g}}). \end{aligned}$$

Newton's method with line search is applied to solve (17) and a local minimizer is guaranteed to be found [13].

The method to solve subproblem (14) is summarized as follows:

Subalgorithm: two dimensional subspace method

1. Given any initial feasible point \mathbf{x}_0 and the maximum iteration number T^{inner} , $t_1 = 1$.
2. Calculate $\tilde{\mathbf{g}}$ and construct subproblem (17).
3. Solve (17) by Newton's method with line search and $\mathbf{x}_0 := \sqrt{1 - a^2} \mathbf{x}_0 + a \tilde{\mathbf{g}}$. If the objective function of (14) does not reduce or $t_1 = T^{\text{inner}}$, stop and output \mathbf{x}_0 ; else $t_1 := t_1 + 1$, go to Step 2.

The maximum iteration number T^{inner} relates to the efficiency of the algorithm. It should not be too large or too small, in order to avoid spending too much time for subproblem or insufficient reduction of the objective function value. We will discuss its specific choice in Section "Simulations".

As the projection to the Stiefel manifold $\mathbf{x}^H \mathbf{x} = 1$ is very easy to compute, projected gradient method is also a common approach to solve (14). In the following theorem and remark, we compare our 2D subspace method with the projected gradient method and show the advantage of our method.

Theorem 1 Denote the projected gradient step started from \mathbf{x}_0 with stepsize α as:

$$\mathbf{x}(\alpha) = \frac{\mathbf{x}_0 - \alpha \mathbf{g}_0}{\|\mathbf{x}_0 - \alpha \mathbf{g}_0\|}. \quad (18)$$

On one hand, for any vector $\mathbf{x} \in S(\mathbf{x}_0, \tilde{\mathbf{g}})$, there exists a stepsize $\alpha \in \mathbb{R}$, such that \mathbf{x} is expressed as (18); on the other hand, for any $\alpha \in \mathbb{R}$, the projected gradient step $\mathbf{x}(\alpha)$ can be expressed as the linear combination of \mathbf{x}_0 and $\tilde{\mathbf{g}}$, which means $\mathbf{x}(\alpha) \in S(\mathbf{x}_0, \tilde{\mathbf{g}})$.

Proof. See the Appendix.

Remark 1 Theorem 1 shows that step in certain 2D subspace and the projected gradient step can be mutually expressed. Thus the minimization of the objective function f on the 2D subspace $S(\mathbf{x}_0, \tilde{\mathbf{g}})$ and that along the projected gradient step are essentially the same. The 2D subspace problem is easy to handle, as shown above; but the search for (18) may be quite difficult, because the projection operator would make the objective function nondifferentiable and lose the common Taylor expansion, consequently it is nontrivial to search for proper stepsize α . Due to these considerations, we use the 2D subspace method rather than the projected gradient method to solve (14).

Framework of AHT algorithm

According to the above discussions, the framework of the AHT algorithm can be concluded as follows (Algorithm 2).

Algorithm 2: algorithm with Householder transformation

1. Set initial precoders and decoders as $\mathbf{V}_k^0, \mathbf{U}_k^0, k \in \mathcal{K}$, parameter T^{inner} in 2D subspace, the maximum inner iteration number T^{outer} , the initial penalty parameter C , parameter $\sigma_0 > 1$ for C and the stopping parameter ε . $i = 0$.
2. Solve (11) iteratively: fix $\mathbf{U}_k^i, \mathbf{V}_k^i, k \in \mathcal{K}$, and set the initial normalized vectors $\mathbf{u}_k, \mathbf{v}_k, k \in \mathcal{K}$. $t = 1$.
 - (2.1) For precoders: fix $\mathbf{u}_k, k \in \mathcal{K}$. For $k \in \mathcal{K}$, construct $\mathbf{A}, \mathbf{B}, \mathbf{C}$ in (14) and solve it with 2D subspace method to obtain \mathbf{v}_k .
 - (2.2) For decoders: fix $\mathbf{v}_k, k \in \mathcal{K}$. Similarly to Step 2.1, obtain new $\mathbf{u}_k, k \in \mathcal{K}$ by solving (14).
 - (2.3) Calculate $\mathbf{U}_k^{i+1}, \mathbf{V}_k^{i+1}, k \in \mathcal{K}$ from (10). If $t = T^{\text{outer}}$ or $P^l(\mathbf{U}_k^{i+1}, \mathbf{V}_k^{i+1}) \leq P^l(\mathbf{U}_k^i, \mathbf{V}_k^i) - 1/C$, go to Step 3; else $t := t + 1$, go to step 2.1.
3. If $P^l(\mathbf{U}_k^{i+1}, \mathbf{V}_k^{i+1}) > P^l(\mathbf{U}_k^i, \mathbf{V}_k^i) - 1/C$, increase the penalty parameter $C := \sigma_0 C$. If $P^l(\mathbf{U}_k^{i+1}, \mathbf{V}_k^{i+1}) < \varepsilon$, stop and output $\mathbf{U}_k := \mathbf{U}_k^{i+1}, \mathbf{V}_k := \mathbf{V}_k^{i+1}, k \in \mathcal{K}$; else $i := i + 1$, go to Step 2.

In our simulations, t is usually no more than 5 when $P^l(\mathbf{U}_k^{i+1}, \mathbf{V}_k^{i+1}) \leq P^l(\mathbf{U}_k^i, \mathbf{V}_k^i) - 1/C$ is achieved in Step 2. T^{outer} , the maximum number of inner iterations inside Step 2, acts to avoid extreme situation with too much time to solve subproblems. Thus we suggest $T^{\text{outer}} = 5$.

Hybrid algorithm

Although the AHT algorithm enjoys lower complexity than MAMA as analyzed in the next section, it solves subproblem (14) inexactly by successive 2D subspace minimizations. This reduces the objective function value rapidly at first, however the speed becomes much more slowly when the iterative point approaches a local optimal solution. Simulation also verifies that it is quite difficult to align the interference perfectly with the AHT algorithm. In contrast, MAMA solves subproblem (9) exactly with closed form solution. The sequence generated by MAMA converges fast near the solution, which is complementary to that of AHT.

In order to make full use of the low complexity of AHT and the fast local convergence property of MAMA, it seems reasonable for us to combine these two methods together into a hybrid algorithm. We believe that such a hybrid algorithm will enjoy the properties of low complexity and fast local convergence, which are verified by our simulation tests reported in Section ‘‘Simulations’’.

This hybrid algorithm consists of two stages. In the first stage, it uses the AHT algorithm to iterate from any starting point to a point near the solution. Then, the algorithm switches to MAMA, namely it applies MAMA to converge to the solution, by starting from the point obtained by AHT (Algorithm 3).

Algorithm 3: hybrid algorithm

1. Set initial precoders and decoders as $\mathbf{V}_k^0, \mathbf{U}_k^0, k \in \mathcal{K}$, the penalty parameter $C = 1$.
2. Input the initial iterative point and penalty parameter into the AHT algorithm. Set its stopping criterion as $P^l(\mathbf{U}_k, \mathbf{V}_k) < 0.01P^l(\mathbf{U}_k^0, \mathbf{V}_k^0)$. Output the corresponding $\mathbf{V}_k, \mathbf{U}_k, k \in \mathcal{K}$ and the current penalty parameter $C_{\text{AHT}} = C$.
3. Input $\mathbf{V}_k, \mathbf{U}_k, k \in \mathcal{K}$ and $C = C_{\text{AHT}}$ as the initial iterative point and penalty parameter of MAMA, respectively. Iterate until convergence by the MAMA. Output the corresponding precoders and decoders $\mathbf{V}_k, \mathbf{U}_k, k \in \mathcal{K}$.

In some sense, it can be viewed that the AHT algorithm acts as a to rapidly find a good initial point with low interference for MAMA. Starting from the good initial point, MAMA can converge very fast to a local optimal solution with perfect IA.

Analysis of computational complexity

In this section, we analyze the computational complexity of IA algorithms for PM model, including our two proposed algorithms MAMA and AHT, as well as the MSP algorithm from [11].

Advantages of low complexity algorithms

Before presenting the detailed analysis on the computational complexities of different algorithms, we provide the reasons for constructing low computational complexity IA algorithms for the PM model.

1. No matter our algorithms or the existed IA algorithms, they all exploit the channel reciprocity to design the transmit precoders and receiver decoders in a distributed way. Such reciprocity is based on the time division duplex (TDD) operation with synchronized time-slot in the practical system. However, due to the time varying property of wireless channel, perfect reciprocity may be hard to achieve and result in residual interference at the receivers. This requires algorithms with short computing time to reduce the possibility of performance loss due to the imperfect reciprocity in practical systems [15].
2. Furthermore, that the receivers generally have limited computational complexity, might be a bottleneck for the complexity of the algorithms. Thus we need to design algorithms with low complexity and simple computation, to satisfy the computation restriction of communication equipments.
3. Also, algorithms with lower complexity are easier to be extended to large scale problems. For small scale problems, algorithms with low or high complexity do not differ much from each other. However as the problem dimension grows, high complexity algorithms can be very very slow for large scale problems because of too much computing time, while algorithms with lower complexity are more preferable.

Detailed analysis

Here we consider the number of complex multiplications as the complexity criterion. The AHT, MAMA and MSP algorithm are analyzed and compared. Their main computations in one iteration are listed as follows:

1. The computation of $\mathbf{H}_{kl}\mathbf{V}_l\mathbf{V}_l^H\mathbf{H}_{kl}^H$ and $\mathbf{H}_{lk}^H\mathbf{U}_l\mathbf{U}_l^H\mathbf{H}_{lk}$, $k, l \in \mathcal{K}$ are required for all the three algorithms. Let $\tilde{\mathbf{H}}_{kl} = \mathbf{H}_{kl}\mathbf{V}_l$. Then we can calculate $\mathbf{H}_{kl}\mathbf{V}_l\mathbf{V}_l^H\mathbf{H}_{kl}^H$ as $\tilde{\mathbf{H}}_{kl}\tilde{\mathbf{H}}_{kl}^H$. Similarly we can compute $\mathbf{H}_{lk}^H\mathbf{U}_l\mathbf{U}_l^H\mathbf{H}_{lk}$ by introducing $\tilde{\mathbf{H}}_{lk} = \mathbf{H}_{lk}^H\mathbf{U}_l$. As k and l traverse all the elements in \mathcal{K} , the entire complexity is $K^2[(dMN + dN^2) + (dMN + dM^2)] = K^2d(2MN + N^2 + M^2)$.
2. Besides term 1, the AHT algorithm requires to compute $\mathbf{U}_k\mathbf{U}_k^H$ and $\mathbf{V}_k\mathbf{V}_k^H$, with complexity of $Kd(M^2 + N^2)$.
3. P^l is computed in both the AHT and MAMA. Noticing that

$$P^l = \sum_{k \in \mathcal{K}} \sum_{l \neq k, l \in \mathcal{K}} \|\mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l\|_F^2 = \sum_{k \in \mathcal{K}} \sum_{l \neq k, l \in \mathcal{K}} \|\mathbf{V}_k^H \mathbf{H}_{lk}^H \mathbf{U}_l\|_F^2,$$

we can compute $\|\mathbf{U}_k^H \tilde{\mathbf{H}}_{kl}\|_F^2$ if $N < M$, or $\|\mathbf{V}_k^H \tilde{\mathbf{H}}_{lk}\|_F^2$ otherwise, where $\tilde{\mathbf{H}}_{kl}$ and $\tilde{\mathbf{H}}_{lk}$ are the pre-computed parameters in term 1. With the above analysis, the corresponding complexity is $K(K-1)d^2(\min(M, N) + 1)$.

4. AHT algorithm requires to update precoders and decoders by computing $\mathbf{U}_k^{i+1}, \mathbf{V}_k^{i+1}, k \in \mathcal{K}$ in (10). With $\mathbf{U}_k^{i+1} = \mathbf{U}_k^i - 2\mathbf{u}_k(\mathbf{u}_k^H \mathbf{U}_k^i)$, the complexity is $K[(d^2 + dM) + (d^2 + dN)] = Kd(2d + M + N)$.
5. In each inner iteration of our 2D subspace method for (14) in the AHT algorithm, it mainly requires computation of $\mathbf{Ax}, \mathbf{Bx}, \mathbf{Cx} = [\mathbf{A}(\mathbf{Bx}) + \mathbf{B}(\mathbf{Ax})]/2, \mathbf{Ag}$ and \mathbf{Bg} , the complexity of which is $6n^2$, where n is the dimension of \mathbf{x} . Suppose for each 2D subspace method, there are mostly T^{inner} inner iterations, we can see that each iteration of AHT requires the total complexity of $6KT^{\text{inner}}(M^2 + N^2)$.
6. One eigenvalue decomposition of matrix with dimension $M \times M$ requires $9M^3$ complexity [16]. Both the MAMA and MSP algorithm need K eigenvalue decompositions of $M \times M$ matrices and K with dimension $N \times N$. The entire complexity is $9K(M^3 + N^3)$.
7. The complexity of one compact SVD of matrix with dimension $M \times d$ ($d < M$) is $14Md^2 + 8d^3$ [16]. The MSP algorithm requires K compact SVDs for matrices with dimension $M \times d$ and K with dimension $N \times d$, whose complexity is $Kd^2(14M + 14N + 16d)$.

Table 1 shows the specific computational complexity of the three algorithms. Comparing the AHT and MAMA, besides the common terms, AHT owns term 2,4,5, while MAMA has term 6. With the fact that $1 \leq d \leq \min(M, N)$, we can deduce that:

$$\begin{aligned} Kd(2d + M + N) + Kd(M^2 + N^2) &\leq 2Kd(M + N) + Kd(M^2 + N^2) \\ &\leq 3K(M^2 + N^2) < 3K(M^3 + N^3). \end{aligned} \quad (19)$$

For term 5, as long as $T^{\text{inner}} \leq \min(M, N)$, we have

$$6KT^{\text{inner}}(M^2 + N^2) \leq 6K(M^3 + N^3). \quad (20)$$

Adding both sides of (19) and (20), we conclude that AHT has lower complexity than MAMA.

Table 1 Comparison of computational complexity

| | AHT | MAMA | MSP |
|--|-----|------|-----|
| Matrix multiplication: $K^2d(2MN + N^2 + M^2)$ | ✓ | ✓ | ✓ |
| $\mathbf{U}_k \mathbf{U}_k^H, \mathbf{V}_k \mathbf{V}_k^H, k \in \mathcal{K}: Kd(M^2 + N^2)$ | ✓ | | |
| $P^l: K(K-1)d^2(\min(M, N) + 1)$ | ✓ | ✓ | |
| $\tilde{\mathbf{U}}_k, \tilde{\mathbf{V}}_k, k \in \mathcal{K}: Kd(2d + M + N)$ | ✓ | | |
| Subproblem construction: $6KT^{\text{inner}}(M^2 + N^2)$ | ✓ | | |
| Eigenvalue decomposition: $9K(M^3 + N^3)$ | | ✓ | ✓ |
| Compact SVD: $Kd^2(14M + 14N + 16d)$ | | | ✓ |

Similarly, we compare term 3 and term 7 to see the difference between the complexity of MAMA and MSP. As long as $K \leq 23$ (which is usually the case in practical considered IA problems), the following inequality holds:

$$\begin{aligned} K(K-1)d^2(\min(M, N) + 1) &\leq 22Kd^2(\min(M, N) + 1) \\ &\leq Kd^2(22\min(M, N) + 22d) \leq Kd^2(14M + 14N + 16d). \end{aligned}$$

So the complexity of MAMA is lower than that of the MSP algorithm. Thus the algorithms ranked from low to high complexity in one iteration are AHT, MAMA and MSP. This also implies that the hybrid algorithm has lower complexity than MAMA and MSP.

Our above analysis compares the complexity of each iteration of different algorithms. However, in order to compare the computational complexity of different algorithms, we need to estimate the total number of iterations of all the algorithms under considerations, which is not easy. Therefore, we try to explain it by convergence curves with consuming time in simulations.

Simulations

In this section, we analyze our proposed algorithms the MAMA, AHT and hybrid algorithm by simulations, and compare them with the existed AMA [7] and MSP algorithm proposed in [11]. $(5 \times 5, 2)^4$ interference channels are considered, that is, $K = 4, M = N = 5, d = 2$, which satisfy the general feasibility condition [2, 3]. Each component of $\mathbf{H}_{kl}, k, l \in \mathcal{K}$ is i.i.d complex Gaussian distribution $\mathcal{CN}(0, 1)$.

We use sum rate as the measure of quality of service. The sum rate of the K -user MIMO interference channel is expressed as follows:

$$R = \sum_{k \in \mathcal{K}} \log \left| \mathbf{I}_N + \mathbf{F}_k^{-1} \mathbf{L}_{kk} \right| = \sum_{k \in \mathcal{K}} \log \left| \mathbf{I}_N + \left(\sigma^2 \mathbf{I}_N + \sum_{l \neq k, l \in \mathcal{K}} \mathbf{L}_{kl} \right)^{-1} \mathbf{L}_{kk} \right|, \quad (21)$$

where $\mathbf{L}_{kl} = \mathbf{H}_{kl} \mathbf{V}_l \mathbf{V}_l^H \mathbf{H}_{kl}^H, k, l \in \mathcal{K}, \mathbf{F}_k = \sigma^2 \mathbf{I}_N + \sum_{l \neq k, l \in \mathcal{K}} \mathbf{L}_{kl}$ and σ^2 is the covariance of the additive white Gaussian noise at the receivers. Here we define the SNR as $\text{SNR} = P/\sigma^2 = d/\sigma^2$.

In the AHT algorithm, we set $T^{\text{outer}} = 5$ and initially $C = 1$. For each figure, 250 random realization of different channel coefficients are generated to evaluate the average performance. For each realization the initial values of precoders and decoders $\mathbf{V}_k^0, \mathbf{U}_k^0, k \in \mathcal{K}$ are randomly generated and remain the same in the compared algorithms.

Parameter analysis

As mentioned before, the choice of T^{inner} relates to the efficiency of the AHT algorithm and also the hybrid algorithm. In Figure 2 we plot the convergence curves of the hybrid algorithms with parameter $T^{\text{inner}} = 3, 5, 10$, respectively. SNR equals 30 dB and $\sigma_0 = 20$ here.

Figure 2 Convergence of the average sum rate of the hybrid algorithms with $T^{\text{inner}} = 3, 5, 10$.

Figure 2 reveals that, both the converged average sum rate and the convergence rate of the hybrid algorithms with different T^{inner} are quite similar. The curve with $T^{\text{inner}} = 10$ seems to be the worst, which accords with our complexity analysis as $T^{\text{inner}} > \min(M, N) = 5$. In the upcoming simulations we set $T^{\text{inner}} = 5$ with which the hybrid algorithm obtains the highest sum rate, to improve the efficiency of our proposed algorithms.

Besides T^{inner} , we also compare the performances of the hybrid algorithms with different σ_0 . σ_0 determines the growth speed of the penalty parameter C , which represents the weight of P^l in the objective function of (7). If σ_0 is too small, it would take many iterations for C to grow large, and thus affects the efficiency of the hybrid algorithm. On the other side, too large σ_0 forces C to grow too fast, which makes (7) degenerate into the LIM problem in short time and that P^S cannot be increased efficiently.

Based on the aforementioned analysis, we plot the convergence curves of the hybrid algorithms with $\sigma_0 = 10, 20, 50$ in Figure 3. SNR is also set as 30 dB here. The three curves are very close to each other, which indicates that the algorithm is not sensitive to σ_0 . In the following simulations we set $\sigma_0 = 20$.

Figure 3 Convergence of the average sum rate of the hybrid algorithms with $\sigma_0 = 10, 20, 50$

AHT algorithm

In this section, the relative iteration number to achieve certain leakage interference of the AHT and MSP algorithm are displayed in Figure 4, to show the property of AHT algorithm. The stopping criterion is $P^l \leq 0.01$. The average number of iterations of the MSP and AHT algorithm to solve one problem are 1233 and 583, respectively. To analyze the relative convergence performance, we use the normalized iterations, the ratio of the iterations at the certain point and the total iterations, as the x -axis. And the y -axis represents the total leakage interference P^l . Figure 4 implies that to achieve the interference of 0.5 from the initial point, the AHT algorithm requires 7.5% of its total iterations while MSP requires 45%. Compared to the MSP algorithm, the AHT algorithm iterates rapidly to a point with low leakage interference P^l . Actually the main consuming time of AHT

algorithm is used to reduce P^I from less than 1 towards 0.01, and the convergence rate becomes much slower when P^I is smaller. The performances here accords with the previous analysis.

Figure 4 Comparison of AHT and MSP: total leakage interference versus normalized iteration number

Comparison of different IA algorithms

In this section, the performances of different IA algorithms are compared and analyzed, including our proposed MAMA, AHT and hybrid algorithm, as well as the existed MSP and AMA. The number of iterations of each algorithm to solve each problem is set as 3000, in order to perfectly align the interference. To take full advantage of AHT, we set its stopping parameter $\varepsilon = 0.01P^I(\mathbf{U}_k^0, \mathbf{V}_k^0)$, where $\mathbf{V}_k^0, \mathbf{U}_k^0, k \in \mathcal{K}$ are the initial precoders and decoders.

The average sum rate and relative computing time of the four algorithms (MAMA, Hybrid, MSP and AMA) with respect to different SNRs are depicted in Figures 5 and 6, respectively. Both our proposed algorithms show good performances, especially the hybrid algorithm. As reflected in Figure 5, the curves representing the average sum rate of MSP, MAMA and the hybrid algorithms are quite close to each other. The solutions of the three algorithms are quite different, whereas they get similar sum rate under the same SNR. And all the three algorithms gain about 5 bps/Hz higher sum rate than the AMA under medium and high SNR. In Figure 6, both of our proposed two algorithms require much less computing time than the MSP algorithm. Particularly, the hybrid algorithm gain almost as high sum rate as MSP with as little time as the AMA.

Figure 5 Comparison of MAMA, Hybrid, MSP and AMA: average sum rate versus SNR

Figure 6 Comparison of MAMA, Hybrid, MSP and AMA: average relative computing time versus SNR.
Time of AMA is set as standard

To further compare the convergence performances of these four algorithms as well as AHT algorithm, we plot their convergence curves of sum rate with respect to the running time in Figure 7. As AHT algorithm is difficult to converge and eliminate interference, it is shown as the first stage of the hybrid algorithm. Its output point is used to continue iteration in the hybrid algorithm. SNR are set as 5, 15 and 30 dB to represent the scenarios of low, medium and high SNR, respectively. In all three scenarios, both MAMA and the hybrid algorithm achieve similar converged average sum rate as the MSP algorithm, which gain more sum rate than the AMA. To achieve a certain sum rate, MAMA consumes the least time. We also observe an interesting phenomenon, that in the low and medium SNR scenario each of the MAMA, hybrid and MSP algorithm achieves high sum rate during iteration before convergence, after that it reduces and converges to a lower rate. This may be relevant to the value of σ^2 . As shown at the beginning of this section, SNR value is inversely proportional to σ^2 . For the low SNR scenario, σ^2 is considerably high and $\sigma^2\mathbf{I}_N$ takes the main part of \mathbf{F}_k in (21). Then, the sum rate mainly increases with large P^S . All the three concerned algorithms are designed to increase P^S in the first few iterations and gradually reduce the weight of this requirement. This may lead to the phenomenon that P^S increases to a peak value and then decreases gradually, which explains the similar phenomenon of sum rate in Figure 7.

Figure 7 Convergence of the average sum rate of MAMA, AHT, Hybrid, MSP and AMA with respect to SNR = 5, 15, 30 dB

Similar as Figure 7, for the five compared algorithms (MSP, AHT, Hybrid, MAMA and AMA), we also plot the convergence curves of the leakage interference with respect to the running time in Figure 8. As the convergence curves of the same algorithm are quite close for different SNR scenarios, here we only display the scenario that SNR is 30 dB for representation. Among the algorithms, the hybrid algorithm converges fastest to a point with interference less than 10^{-4} . The MAMA also converges faster than the MSP and AMA.

Figure 8 Convergence of the leakage interference of MAMA, AHT, Hybrid, MSP and AMA with respect to SNR = 30 dB

Comparing the proposed MAMA and hybrid algorithm from both aspects of computing time and convergence rate in Figures 6, 7 and 8, we should admit that the improvement of the hybrid algorithm over MAMA is limited in the examples that we tested. The main reason of this phenomenon is due to the quite small scale of the test problems, for which lower complexity algorithm does not have much gain. Nevertheless, as the dimension of the problem increases, the hybrid algorithm will save more consuming time and benefit more. Moreover, in similar applications in other fields, there might be large scale similar matrix optimization problems, we believe that the hybrid algorithm will improve MAMA greatly.

Remark 2 *We have pointed out in Section “A hybrid algorithm” that the sequence generated by the hybrid algorithm converges to a local optimal solution. Although it is not guaranteed to be global optimal, the obtained local optimal solution performs with high sum rate (almost the same as MSP algorithm and higher than the AMA, as depicted in Figure 5) and perfect IA. Such solution satisfies our requirement for an IA solution. The performances of MAMA and the hybrid algorithm in this section indicates that the Courant penalty function technique is an effective way to combine the leakage interference and the desired signal power together. Through this technique, both algorithms achieve high average sum rate in short time. In the last few iterations when C becomes very large, the algorithm essentially tries to minimize P^I with the part of $-P^S$ being nearly ignored. Therefore, MAMA (also the hybrid algorithm) eventually behaves very similar to the original AMA, which can perfectly align the interference.*

Conclusion

This article proposed two low complexity algorithms for the desired signal power maximization IA problems of MIMO channels. The IA constraints are added to the objective function and combined with the desired signal power via the Courant penalty function technique. First, a MAMA is proposed following the similar approach of the AMA. Then, a hybrid algorithm is proposed to further reduce complexity. In the hybrid algorithm, the AHT is proposed to iterate rapidly from any initial point to precoders and decoders with low leakage interference. This step provides a good point around the local optimal solution. From this point, MAMA is applied to converge fast to the local optimal solution satisfying our requirement. Analysis shows that, among the compared algorithms the hybrid algorithm has the lowest computational complexity, followed by MAMA, with MSP being the highest. Simulations indicate that both the hybrid algorithm and MAMA achieve similar sum rate as the MSP algorithm with less computing time, and higher sum rate than the AMA.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank the editor and two anonymous referees for their comments and suggestions which improved the article greatly. This work was partly supported by the National Natural Science Foundation of China (NSFC) grants 10831006, 11021101, by CAS grant kjcx-yw-s7 and by the Fundamental Research Funds for the Central Universities.

Appendix

Proof of Theorem 1

As stated in (16), \mathbf{g}_0 is the linear combination of \mathbf{x}_0 and $\tilde{\mathbf{g}}$. Suppose

$$\mathbf{g}_0 = c_1 \mathbf{x}_0 + c_2 \tilde{\mathbf{g}},$$

where c_1 and c_2 are scalars.

First prove that the projected gradient step can be expressed by 2D subspace step. For the projected gradient step $\mathbf{x}(\alpha)$ with any $\alpha \in \mathbb{R}$, we would like to find corresponding scalars a and b , such that

$$\mathbf{x}(\alpha) = \frac{\mathbf{x}_0 - \alpha \mathbf{g}_0}{\|\mathbf{x}_0 - \alpha \mathbf{g}_0\|} = \frac{(1 - c_1 \alpha) \mathbf{x}_0 - c_2 \alpha \tilde{\mathbf{g}}}{\|(1 - c_1 \alpha) \mathbf{x}_0 - c_2 \alpha \tilde{\mathbf{g}}\|} = b \mathbf{x}_0 + a \tilde{\mathbf{g}}. \quad (22)$$

Let the coefficients of \mathbf{x}_0 and $\tilde{\mathbf{g}}$ remain unchanged, then

$$a = \frac{-c_2 \alpha}{\|(1 - c_1 \alpha) \mathbf{x}_0 - c_2 \alpha \tilde{\mathbf{g}}\|}, b = \frac{1 - c_1 \alpha}{\|(1 - c_1 \alpha) \mathbf{x}_0 - c_2 \alpha \tilde{\mathbf{g}}\|}.$$

Next show that 2D subspace step can also be expressed by the projected gradient step. For any $\mathbf{x} \in S(\mathbf{x}_0, \tilde{\mathbf{g}})$, we have $\mathbf{x} = b \mathbf{x}_0 + a \tilde{\mathbf{g}}$. As we wish to find $\alpha \in \mathbb{R}$ such that $\mathbf{x} = \mathbf{x}(\alpha)$, then (22) holds. Similar to the proof above, we can induce that

$$\frac{b}{a} = \frac{c_1 \alpha - 1}{c_2 \alpha}, \alpha = \frac{a}{c_1 a - c_2 b}.$$

Thus we have proved Theorem 1.

References

1. VR Cadambe, SA Jafar, Interference alignment and degrees of freedom of the K -user interference channel. *IEEE Trans. Inf. Theory* **54**(8), 3425–3441 (2008)
2. CM Yetis, T Gou, SA Jafar, AH Kayran, On feasibility of interference alignment in MIMO interference networks. *IEEE Trans. Signal Process.* **58**(9), 4771–4782 (2010)
3. M Razaviyayn, G Lyubeznik, ZQ Luo, On the degrees of freedom achievable through interference alignment in a MIMO interference channel, *IEEE Trans. Signal Process.* **60**(2), 812–821 (2012)
4. SW Choi, S-Y Chung, On the multiplexing gain of K -user partially connected interference channel, preprint, <http://arxiv.org/pdf/0806.4737>. Jun 2008
5. M Guillaud, D Gesbert, Interference alignment in the partially connected K -user MIMO interference channel, in *European Signal Processing Conference (EUSIPCO)*. Barcelona, Spain, 2011, pp. 1095–1099.
6. N Lee, D Park, Y-D Kim, Degrees of freedom on the K -user MIMO interference channel with constant channel coefficients for downlink communications, in *Proc. IEEE Global Telecommun. Conf., (GLOBECOM)*, Hawaii, USA, 2009, pp. 1-6
7. K Gomadam, VR Cadambe, SA Jafar, A distributed numerical approach to interference alignment and applications to wireless interference networks. *IEEE Trans. Inf. Theory* **57**(6), 3309–3322 (2011)
8. YF Liu, YH Dai, ZQ Luo, On the complexity of leakage interference minimization for interference alignment, in *IEEE Int. Workshop Signal Process. Advances Wireless Comm.*, San Francisco, USA, 2011, pp. 471–475
9. DA Schmidt, W Utschick, ML Honig, Large system performance of interference alignment in single-beam MIMO networks, in *Proc. IEEE Global Telecommun. Conf., (GLOBECOM)*, Miami, USA, 2010, pp. 1-6
10. H Shen, B Li, MX Tao, XD Wang, MSE-based transceiver designs for the MIMO interference channel. *IEEE Trans. Wirel. Commun.* **9**(11), 3480–3489 (2010)
11. I Santamaria, O Gonzalez, RW Heath Jr, SW Peters, Maximum sum-rate interference alignment algorithms for MIMO channels, in *Proc. IEEE Global Telecommun. Conf., (GLOBECOM)*, Miami, USA, 2010, pp. 1–6
12. J Nocedal, SJ Wright, *Numerical Optimization* (Springer, Berlin, 1999), pp. 488–522
13. YX Yuan, *Computational Method for Nonlinear Optimization (in Chinese)* (Science Press, Beijing, 2008), pp. 16–20; 158–162
14. YX Yuan, in *Some Topics in Industrial and Applied Mathematics (Series in Contemporary Applied Mathematics CAM 8)*, ed. by R Jeltsch, DQ Li, IH Sloan (Higher Education Press, Beijing, 2007), pp. 206–218

15. HG Ghauch, CB Papadias, Interference alignment: a one-sided approach, in *Proc. IEEE Global Telecommun. Conf., (GLOBECOM)*, Houston, USA, 2011, pp. 1–6
16. GH Golub, CF Van Loan, *Matrix Computations*, 3rd edn. (The Johns Hopkins University Press, London, 1996), pp. 248–255; 414–425

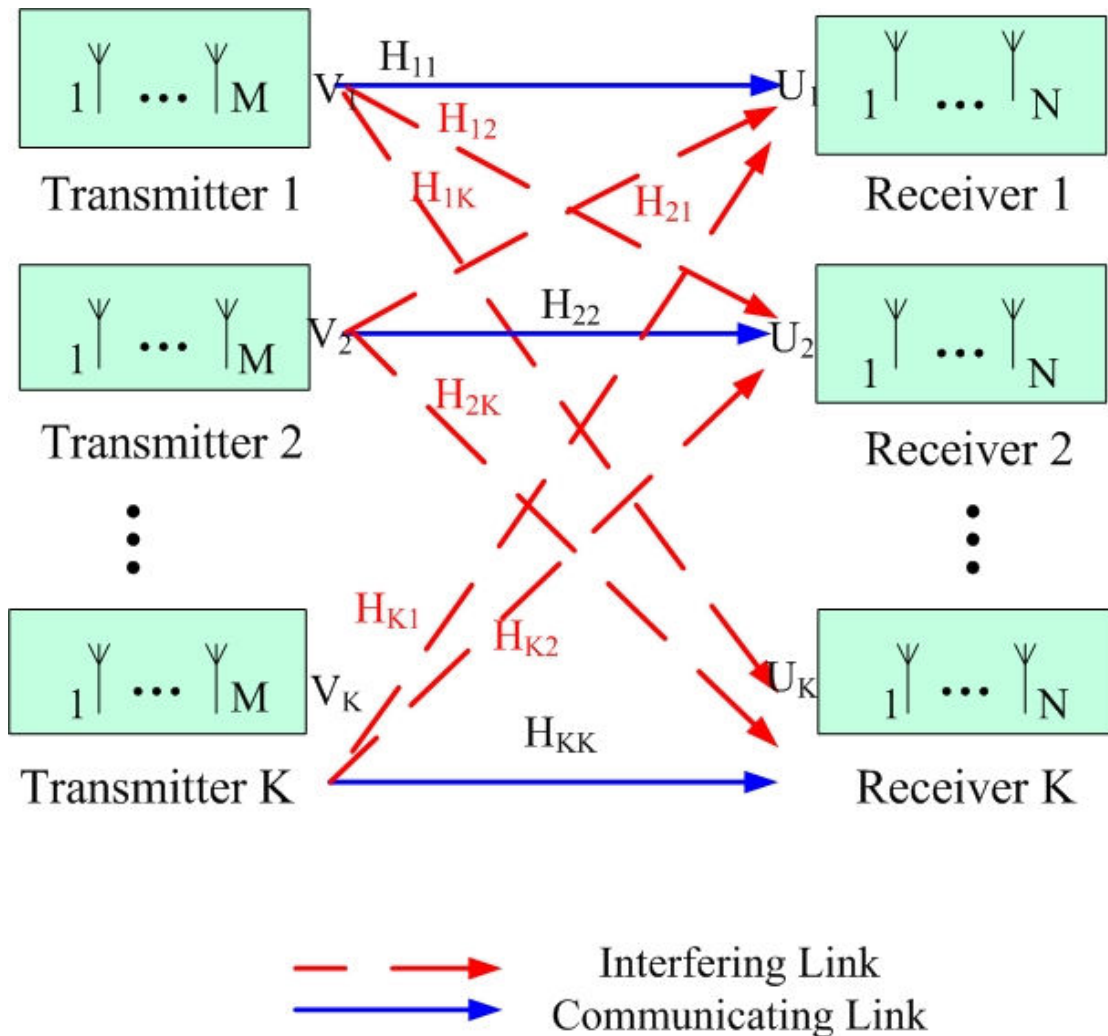


Figure 1

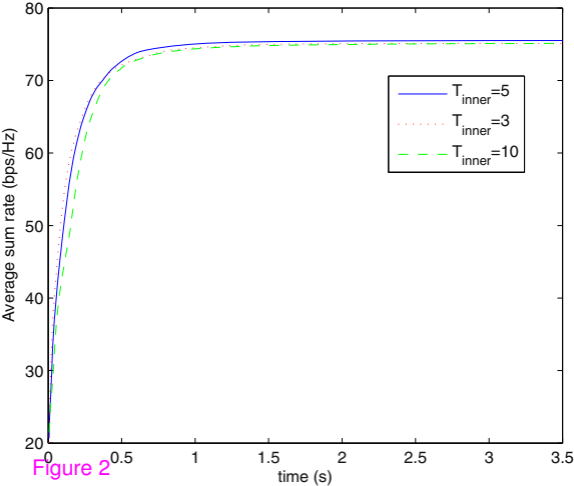


Figure 2

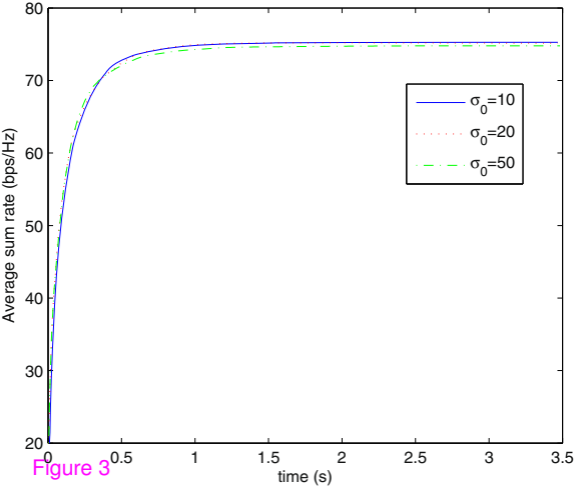
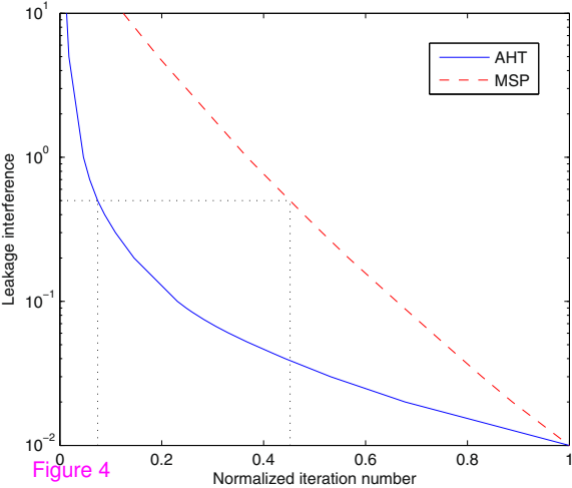


Figure 3



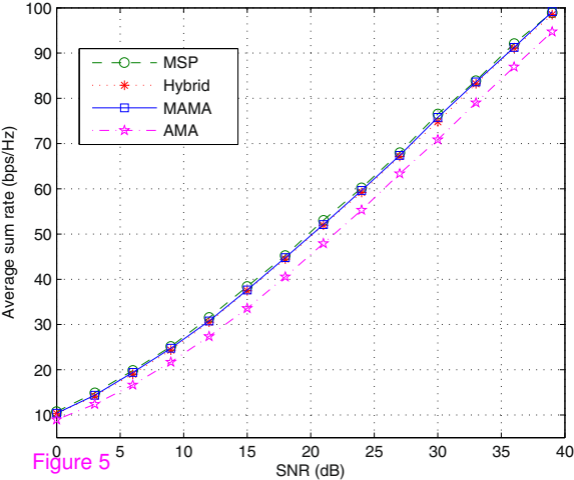


Figure 5

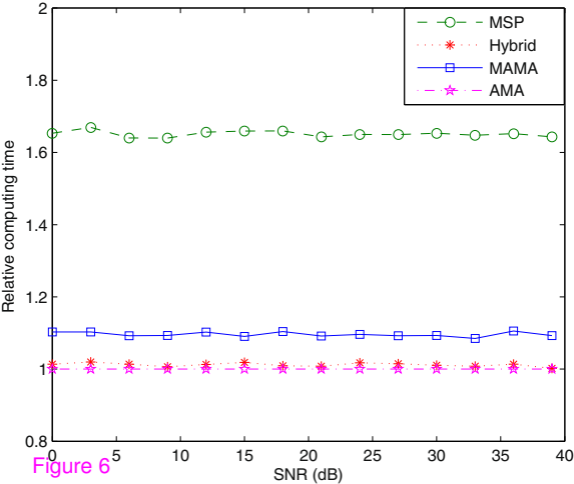


Figure 6

