

RESEARCH

Open Access

Decentralized practical design and centralized benchmark for analog network coding

Enoch Lu*, Zihao You and I-Tai Lu

Abstract

This article proposes both centralized and decentralized design schemes for an analog network coding multiple-input multiple-output system with a relay node between two end nodes. The proposed centralized scheme is called the generalized iterative approach (GIA). It jointly designs the precoders and decoders at the two end nodes, and the processor at the relay to maximize the sum mutual information. Numerical results for the per-node power constraint show the convergence behavior of the GIA and give a performance benchmark for the analog network coding scheme. The proposed decentralized scheme is a practical joint transceiver and signaling design scheme. The keys to its low signaling load are time-division duplex and a symmetric relay processing matrix. The proposed signaling protocol enables the needed information, including channel state information (CSI), to be available at each node. With the needed CSI, a novel symmetric processor design at the relay is developed to maximize an approximate sum mutual information formula (to reduce the signaling loading, the precoders at both end nodes and the noise propagated from the relay are not considered). Employing singular value decomposition transceivers at the two end nodes, it is remarkable that the proposed decentralized approach performs almost as well as the centralized GIA design. It is concluded that the proposed decentralized scheme is a feasible way to implement analog network coding systems.

Keywords: Amplify-and-forward, Analog network coding, Mutual information, MIMO, Precoder, Signaling, TDD, Transceiver design, Two-way relaying

1. Introduction

Analog network coding (e.g., [1-16]), also known as two-way amplify-and-forward relaying, is a bi-directional relaying technique. Its data transmission takes half the time of conventional bi-directional relaying. Moreover, it accomplishes this feat with simpler relay processing than another network coding technique, decode-and-forward network coding [1]. In analog network coding, both end nodes transmit simultaneously to the relay. The relay then processes its received signal and transmits it back to the end nodes. Finally, the end nodes then subtract their self-interference and decode their desired data.

There is no restriction on how many antennas each node needs to have in order for the system to do analog network coding. There are works on all single-antenna nodes (e.g., [2-4]). There are also works where the nodes can have multiple antennas (e.g., [5-16]). The multiple

antennas can aid in multiplexing and/or diversity. Consequently, the rest of this article is about the case when each node has multiple antennas.

When each node has multiple antennas, one obvious and flexible approach is to use multiple-input multiple-output (MIMO) linear processing: a precoding and decoding matrix at each end node and a processing matrix at the relay. However, the design of these five matrices is not trivial. Some papers try to optimize according to some criteria (e.g., [8-11]) while others seek to propose low-complexity heuristics (e.g., [12,13]). Since the optimization problems are highly nonlinear, the solutions have only been derived numerically (or approximately) and are innately sub-optimum. Moreover, they are not implementable unless they are supportable by practical signaling.

Some have realized this need to consider signaling. For example, Panah and Heath Jr [14] and Roemer and Haardt [15] have proposed channel estimation techniques to supplement their designs. However, both works assume that no channel state information (CSI) is

* Correspondence: enoch.school@gmail.com
Department of ECE, Polytechnic Institute of NYU, 6 MetroTech Center,
Brooklyn, NY 11201, USA

available at the relay. In [14], the relay basically just scales and forwards its received signal. In [15], the relay processing does not have to be just a scaling. But, it is independent of the current CSI. There may be some improvement if CSI-dependent relay processing can be enabled.

Therefore, this article proposes a joint, *but decentralized*, transceiver and signaling design^a with the following goal: to have high-performing MIMO linear processing, not just scaling, at each node using only a small amount of signaling. The practical signaling protocol proposed uses time-division duplex (TDD) and enables the relay to estimate its two outgoing channels (i.e., the channels from the relay to the two end nodes). It also enables the two end nodes to estimate the two effective channels between themselves (i.e., the link from one end node through the relay to the other end node and the same link but in the opposite direction). Most importantly, it enables the two end nodes to cancel their self-interference.

To help keep the amount of signaling low, the proposal has the relay design its own symmetric relay processing matrix. The symmetry and the use of TDD together cause the end-to-end channel in one direction to be the transpose of the other direction's, i.e., reciprocity holds for the two effective channels between the two end nodes. (The conjugate-transpose instead of the transpose is used in a lot of literature for representing the channel reciprocity. However, the conjugate-transpose should be used only for time reversal transmissions. Discussions on reciprocity can be found in most electromagnetic textbooks (e.g., see [17,18])). An end node needs to know the effective channel in each direction in order to choose a channel-dependent precoder and decoder. In general, knowing one effective channel does not imply knowing the other thus necessitating signaling both of them. Due to the symmetry and TDD, only one effective channel needs to be signaled here.

Another design decision is to sequentially design the relay processor and the end node processors; the relay designs its processor and then the end nodes design their own. There are no iterations between the processors. The major benefit of this choice is avoiding signaling repeatedly between the nodes. The chosen way to enable this sequential design is to have the relay design its processor ignoring the precoders, decoders, and the noise propagated from the relay. Though there may be some degradation in the sum mutual information, this decoupling substantially reduces the complexity of the relay. In turn, the lower complexity admits a shorter hardware timeline, etc.—reducing the duration of an analog network coding transmission. Two relay designs which comply with the above design choices are a heuristic approach RRANOMAX in [12] and a novel iterative scheme herein developed which seeks to maximize an approximation to the sum mutual information.

To evaluate the performance of the proposed decentralized scheme, a generalized iterative approach (GIA) is developed

for jointly designing the precoders, decoders, and relay processor to maximize the true sum mutual information. The GIA is a centralized scheme where all CSI, noise, and source statistics have to be known at a central processing unit and, in addition, the processed results have to be available at the end and relay nodes. Because of the large amount of signaling load, the GIA is not very practical but can be employed for generating a performance benchmark. Although mutual information maximization for MIMO relay systems has been studied extensively, those for analog network coding systems have only been recently introduced in [6,7]. Both [6,7] are different from the GIA in that they are both based on the assumption that source precoders are given beforehand. Like the GIA, the studies [9,10] are also for the design of the precoders, decoders, and relay processor. However, its criterion is minimum mean squared error (MMSE). To the best of the authors' knowledge, our GIA is the first work on joint precoders, decoders, and relay processors design to maximize the sum mutual information of MIMO analog network coding systems. It is known that the MMSE and mutual information are related for a MIMO channel with Gaussian noise (Equation 22 in [19]). However, that relation is not satisfied for analog network coding systems because of the following two reasons. First, the relay and precoders chosen for the maximum sum mutual information case are different from those for the MMSE case. Second, the noises at the two end nodes (which include the propagation of the noise received at the relay) for the maximum sum mutual information case are also not the same as those for the MMSE case.

The convergence behaviors of the proposed symmetric relay design and the proposed GIA are studied. Their sum mutual information performances are also studied. It is shown numerically that the decentralized approach performs almost as well as the centralized benchmark, the GIA. It is thus concluded that the proposed decentralized approach is a feasible way to implement analog network coding.

The outline of this article is given as follows. Section 2 lays the formulation foundation. Section 3 presents the GIA, a novel centralized scheme for jointly design the precoders, decoders, and relay processor. Section 4 explains the proposed signaling protocol for decentralized designs and gives a joint precoder–decoder design where no additional signaling is required. Following the decentralized signaling protocol, Section 5 proposes a novel design for a symmetric relay processing matrix according to the approximate sum mutual information metric. The RRANOMAX design [12] is also summarized in Section 5. Numerical results are shown in Section 6. Conclusion is made in Section 7.

The notation of this article is as follows: all boldface letters indicate vectors (lower case) or matrices (upper

case). \mathbf{A}' , $\overline{\mathbf{A}}$, \mathbf{A}^* , \mathbf{A}^{-1} , $tr(\mathbf{A})$, $|\mathbf{A}|$, $\|\mathbf{A}\|_F$, and $\langle \mathbf{A} \rangle$ stand for the transpose, conjugate, conjugate transpose, inverse, trace, determinant, Frobenius norm, and expectation of \mathbf{A} , respectively. \mathbf{I}_r denotes the $r \times r$ identity matrix. $\mathbf{0}$ denotes the zero matrix with proper dimension. $\mathbf{A} > \mathbf{0}$ denotes that \mathbf{A} is a positive definite matrix. $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product of \mathbf{A} and \mathbf{B} . $vec(\cdot)$ and $unvec(\cdot)$ are the matrix vectorization operator and the inverse matrix vectorization operator, respectively. $DIAG(\sigma_1, \sigma_2, \dots, \sigma_r)$ is a diagonal matrix where the elements $\{\sigma_1, \sigma_2, \dots, \sigma_r\}$ are put on the main diagonal. $\{\mathbf{A}_k\}$ denotes the set of matrices, $\mathbf{A}_1, \mathbf{A}_2$, etc. $\max(a, b)$ denotes the maximum of real numbers a and b . On the other hand, $\min(a, b)$ denotes the minimum of real numbers a and b . $(\chi)^+ = \max(\chi, 0)$.

2. Analog network coding formulation

This article considers a TDD system with relay node R , equipped with M_R antennas, between two end nodes E_1 and E_2 , equipped with M_1 and M_2 antennas, respectively. Each node works in half-duplex mode, receiving and transmitting data in different time slots. These nodes perform analog network coding, completing a bi-directional communication between the two end nodes in just two time slots (see Figure 1)—note that this does not include the time slots for signaling. The problem will be formulated in the frequency domain. It is assumed that the guard time is larger than the delay spread so that there is no inter-symbol interference. It is also assumed that the duration for the system to complete a bi-directionary communication is much smaller than the coherence time so that the channels are considered stationary within that duration.

Two time slots for data (not signaling) transmission are used in analog network coding. The end nodes E_1 and E_2

broadcast signals $\mathbf{x}_1 = \mathbf{F}_1 \mathbf{s}_1$ and $\mathbf{x}_2 = \mathbf{F}_2 \mathbf{s}_2$, respectively, to the relay in the first time slot. The relay thus receives

$$\mathbf{y} = \mathbf{H}_1 \mathbf{x}_1 + \mathbf{H}_2 \mathbf{x}_2 + \mathbf{w}. \quad (1)$$

The data vector \mathbf{s}_i is a $d_i \times 1$ vector and is precoded by the $M_i \times d_i$ precoder \mathbf{F}_i ; the scalar d_i is the number of data streams from E_i . Each element of \mathbf{s}_i is assumed to be zero-mean and the entire vector \mathbf{s}_i satisfies $\langle \mathbf{s}_i \mathbf{s}_i^* \rangle = \sigma_{s_i}^2 \mathbf{I} > \mathbf{0}$. The $M_R \times M_i$ matrix \mathbf{H}_i ($i = 1, 2$) is the Rayleigh fading channel from E_i to the relay, \mathbf{y} is the $M_R \times 1$ relay received vector, and \mathbf{w} is the $M_R \times 1$ Gaussian noise vector at the relay. The elements of \mathbf{w} are zero mean and satisfy $\langle \mathbf{w} \mathbf{w}^* \rangle = \Phi_{\mathbf{w}} > \mathbf{0}$. In this article, the transmit power of E_i is constrained by requiring

$$\sigma_{s_i}^2 tr\{\mathbf{F}_i \mathbf{F}_i^*\} = p_i. \quad (2)$$

In the second time slot, the relay transmits $\gamma \mathbf{T} \mathbf{y}$ back to each end node. Here, \mathbf{T} is its $M_R \times M_R$ processing matrix. The scalar γ is chosen to ensure the relay power constraint

$$\begin{aligned} \langle (\gamma \mathbf{T} \mathbf{y})^* \gamma \mathbf{T} \mathbf{y} \rangle &= \gamma^2 tr(\mathbf{Q}) = q, \\ \text{with } \mathbf{Q} &\equiv \mathbf{T} \left(\Phi_{\mathbf{w}} + \sum_{j=1}^2 \sigma_{s_j}^2 \mathbf{H}_j \mathbf{F}_j \mathbf{F}_j^* \mathbf{H}_j^* \right) \mathbf{T}^*. \end{aligned} \quad (3)$$

From (3), we observe that the power constraint on the relay processor depends on the precoders $\{\mathbf{F}_i\}$. Due to the channel reciprocity, E_i 's ($i = 1, 2$) received signal vector (of dimension $M_i \times 1$) is

$$\mathbf{z}_i = \mathbf{H}'_i \gamma \mathbf{T} \mathbf{H}_i \mathbf{x}_i + \mathbf{H}'_i \gamma \mathbf{T} \mathbf{H}_j \mathbf{x}_j + \mathbf{H}'_i \gamma \mathbf{T} \mathbf{w} + \mathbf{a}_i \quad (4)$$

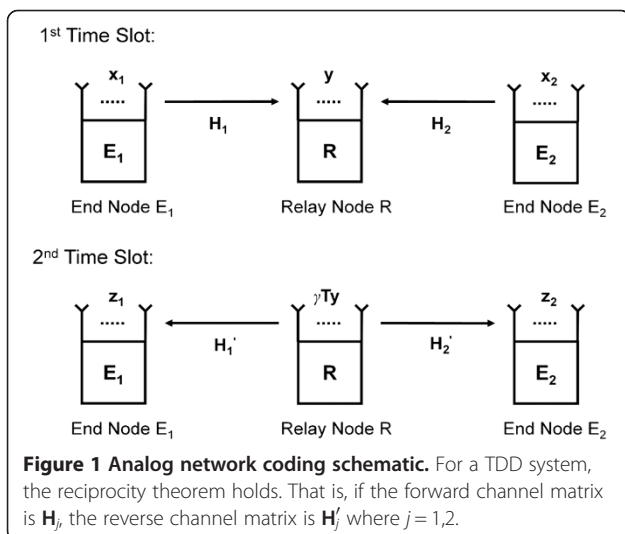
($j = 1$ if $i = 2$ and $j = 2$ if $i = 1$). In the above, \mathbf{a}_i is the $M_i \times 1$ Gaussian noise vector at E_i . It has elements with zero mean and satisfies $\langle \mathbf{a}_i \mathbf{a}_i^* \rangle = \Phi_{\mathbf{a}_i} > \mathbf{0}$. The vectors, $\mathbf{s}_1, \mathbf{s}_2, \mathbf{w}, \mathbf{a}_1$, and \mathbf{a}_2 , are all independent of each other. Also, the channel matrix from the relay to E_i is \mathbf{H}'_i . For convenience, let $\mathbf{C}_{ij} = \mathbf{H}'_i \gamma \mathbf{T} \mathbf{H}_j$ denote the effective channel from the end node E_j , through the relay, to the end node E_i , and let $\mathbf{n}_i = \mathbf{H}'_i \gamma \mathbf{T} \mathbf{w} + \mathbf{a}_i$ represent the effective noise at the end node E_i , where $i, j = 1, 2$. Equation (4) can thus be rewritten as

$$\mathbf{z}_i = \mathbf{C}_{ii} \mathbf{x}_i + \mathbf{C}_{ij} \mathbf{x}_j + \mathbf{n}_i, \quad i \neq j, \quad i, j = 1, 2, \quad (5)$$

where the effective noise covariance matrix is

$$\Phi_{\mathbf{n}_i} = \langle \mathbf{n}_i \mathbf{n}_i^* \rangle = \gamma^2 \mathbf{H}'_i \mathbf{T} \Phi_{\mathbf{w}} \mathbf{T}^* \bar{\mathbf{H}}_i + \Phi_{\mathbf{a}_i}. \quad (6)$$

Look at E_i 's ($i = 1, 2$) received signal vector. It is clear that using only two time slots has caused \mathbf{x}_i and \mathbf{x}_j to be



mixed together in \mathbf{z}_i . That is, E_i interferes with itself. *Somehow* though, E_i knows \mathbf{C}_{ii} , the effective channel from itself, through the relay and back to itself. Since E_i transmitted \mathbf{x}_i , E_i knows \mathbf{x}_i as well and thus can subtract its self-interference completely from \mathbf{z}_i . Consequently, it obtains

$$\hat{\mathbf{z}}_i = \mathbf{z}_i - \mathbf{C}_{ii}\mathbf{x}_i = \mathbf{C}_{ij}\mathbf{x}_j + \mathbf{n}_i, \quad i \neq j, \quad i, j = 1, 2. \quad (7)$$

To decode \mathbf{s}_j , E_i now has the options of regular point to point transmissions (e.g., applying a $d_j \times M_i$ decoder \mathbf{G}_i).

3. Centralized design

Let \mathbf{G}_i denote the decoder at E_i . Then, from [20], the mutual information pertaining to the transmission of \mathbf{s}_j from E_j to E_i is

$$I_{Ei} = \log_2 \frac{|\mathbf{G}_i (\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i}) \mathbf{G}_i^*|}{|\mathbf{G}_i \Phi_{\mathbf{n}i} \mathbf{G}_i^*|}, \quad i, j = 1, 2 \text{ and } i \neq j. \quad (8)$$

Assuming all CSI, source and noise statistics are known at a centralized processing unit, we are to maximize $I_{E1} + I_{E2}$ by jointly designing the decoders $\{\mathbf{G}_1, \mathbf{G}_2\}$, precoders $\{\mathbf{F}_1, \mathbf{F}_2\}$ and relay processor \mathbf{T} subject to the per-node power constraints at the precoders (see (2)) and at the relay (see (3)).

Let $\{\mathbf{F}_1, \mathbf{F}_2, \mathbf{T}\}$ be known. Also let $|\mathbf{G}_{i,\text{OPT}} \Phi_{\mathbf{n}i} \mathbf{G}_{i,\text{OPT}}^*| \neq 0$ where

$$\mathbf{G}_{i,\text{OPT}} = \mathbf{\Pi} \mathbf{F}_j^* \mathbf{C}_{ij}^* \Phi_{\mathbf{n}i}^{-1} \quad (9)$$

and where $\mathbf{\Pi}$ is any invertible square matrix. Note that the commonly used MMSE decoder and singular value decomposition (SVD) decoder (if $\mathbf{F}_j^* \mathbf{C}_{ij}^*$ is invertible) can also be expressed using (9). In this case, Corollary 1 of [20] shows that $\mathbf{G}_{i,\text{OPT}}$ is the optimum decoder and the mutual information I_{Ei} in (8) becomes

$$I_{Ei} = \log_2 \frac{|\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i}|}{|\Phi_{\mathbf{n}i}|}, \quad i, j = 1, 2 \text{ and } i \neq j. \quad (10)$$

Note that $|\mathbf{G}_{i,\text{OPT}} \Phi_{\mathbf{n}i} \mathbf{G}_{i,\text{OPT}}^*| \neq 0$ necessitates that \mathbf{F}_j has full column rank. Due to the way \mathbf{T} is made in this design, γ is not used. The scalar γ is thus equal to 1.

The remaining task now is to choose the precoders $\{\mathbf{F}_1, \mathbf{F}_2\}$ and relay processor \mathbf{T} to maximize the sum mutual information, $I_{E1} + I_{E2}$, subject to the constraints (2–3). That is,

$$\{\mathbf{F}_{1,\text{OPT}}, \mathbf{F}_{2,\text{OPT}}, \mathbf{T}_{\text{OPT}}\} = \arg \max_{\{\mathbf{F}_1, \mathbf{F}_2, \mathbf{T}\}} \{I_{E1} + I_{E2}\} \quad (11)$$

subject to (2–3).

Obtaining an optimum solution to (11) is very difficult due to the intercoupling of the design variables, etc. Consequently, the GIA seeks to find a solution by decoupling the choices for the precoders and relay processor. It does this by iteratively designing them. Before looking at the GIA's actual iterative procedure in Section 3.3, first look at Section 3.1 which gives a way to choose the precoders given the relay processor \mathbf{T} . Then, look at Section 3.2 which gives a way to choose \mathbf{T} given the precoders. The reason is that the findings in Sections 3.1–3.2 are used in Section 3.3.

3.1. Fix relay processor and get precoders

Given a relay processor \mathbf{T} , the cost function in (11), and the constraint (2), this section details one way to choose precoders $\{\mathbf{F}_1, \mathbf{F}_2\}$. As shown in (7), the two-way transmission can be considered as two independent single-user MIMO systems when \mathbf{T} is known. As there exists a closed-form solution for maximizing the mutual information of a single-user MIMO system subject to the per-node power constraint [20], the proposed way is to just apply that closed-form solution here: First, perform the eigenvalue decomposition.

$$\mathbf{C}_{ij}^* \Phi_{\mathbf{n}i}^{-1} \mathbf{C}_{ij} = [\mathbf{V}_j \quad \mathbf{U}_j] \begin{bmatrix} \mathbf{\Xi}_j & \\ & \mathbf{\Gamma}_j \end{bmatrix} [\mathbf{V}_j \quad \mathbf{U}_j]^*, \quad i, j = 1, 2, i \neq j. \quad (12)$$

As is standard, the block matrix $[\mathbf{V}_j \quad \mathbf{U}_j]$ is unitary while $\mathbf{\Xi}_j$ and $\mathbf{\Gamma}_j$ are diagonal matrices with the eigenvalues. As the eigenvalues are arranged in descending order,

$$\mathbf{\Xi}_j = \text{DIAG}(\xi_{j,1}, \xi_{j,2}, \dots, \xi_{j,r_j}) \quad (13)$$

has the r_j largest eigenvalues and $\mathbf{\Gamma}_j$ has the remaining $(M_j - r_j)$ eigenvalues; $r_j = \min(d_j, \text{rank}(\mathbf{C}_{ij}))$. Note that \mathbf{V}_j is $M_j \times r_j$ and \mathbf{U}_j is $M_j \times (M_j - r_j)$. Then, set

$$\mathbf{F}_j = \begin{bmatrix} \mathbf{V}_j \mathbf{\Phi}_j & \mathbf{0}_{M_j \times (d_j - r_j)} \end{bmatrix}, \quad (14)$$

where $\mathbf{\Phi}_j$ is a diagonal matrix with the l th diagonal entry

$$\phi_{j,ll} = \sqrt{\left(\frac{p_j + \sum_{k=1}^{r_j} \xi_{j,k}^{-1}}{\tau \sigma_{s_j}^2} - \frac{1}{\xi_{j,l} \sigma_{s_j}^2} \right)^+} \quad (15)$$

In (15), τ is the number of positive $\phi_{j,l}$ (see Lemma 2 in [20]) and, σ_{sj}^2 and p_j are defined in (2).

3.2. Fix precoders and get relay processor

Given precoders $\{\mathbf{F}_1, \mathbf{F}_2\}$, the cost function in (11), and the constraint in (3), this section details one way to choose relay processor \mathbf{T} . The method of Lagrange multipliers can be used to set up the augmented cost function

$$\xi(\mathbf{T}) = I_{E1} + I_{E2} + \lambda_{\mathbf{T}}(\text{tr}(\mathbf{Q}) - q) \quad (16)$$

where $\lambda_{\mathbf{T}}$ is an unknown Lagrange multiplier. In (16), matrix \mathbf{T} is what we are looking for in design while $\lambda_{\mathbf{T}}$ is solved for so that \mathbf{T} can satisfy its corresponding power constraint. Setting the gradient of the augmented cost function in (16) with respect to \mathbf{T} equal to zero, we obtain (see Appendix 1)

$$\mathbf{X}_1 \mathbf{T}^* \mathbf{Y}_1 + \mathbf{X}_2 \mathbf{T}^* \mathbf{Y}_2 = \mathbf{Z}, \quad (17)$$

where

$$\mathbf{X}_1 = \sum_{k=1}^2 \sigma_{sk}^2 \mathbf{H}_k \mathbf{F}_k \mathbf{F}_k^* \mathbf{H}_k^* + \Phi_{\mathbf{w}}, \quad \mathbf{Y}_1 = \lambda_{\mathbf{T}} \mathbf{I}, \quad (17a)$$

$$\mathbf{X}_2 = \frac{1}{\ln 2} \Phi_{\mathbf{w}},$$

$$\mathbf{Y}_2 = \sum_{i=1}^2 \bar{\mathbf{H}}_i \left(\left(\sigma_{sj}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} - \Phi_{\mathbf{n}i}^{-1} \right) \mathbf{H}'_i, \quad (17b)$$

$$\mathbf{Z} = -\frac{1}{\ln 2} \sum_{i=1}^2 \sigma_{sj}^2 \mathbf{H}_j \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* \left(\sigma_{sj}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} \mathbf{H}'_i. \quad (17c)$$

By using Kronecker products we obtain the relay processor from (17) for a given $\lambda_{\mathbf{T}}$:

$$\mathbf{T} = \left(\text{unvec} \left(\left(\sum_{i=1}^2 \mathbf{Y}_i^T \otimes \mathbf{X}_i \right)^{-1} \text{vec}(\mathbf{Z}) \right) \right)^*. \quad (18)$$

Note that $\mathbf{T} \mathbf{X}_1 \mathbf{T}^* = \mathbf{Q}$ in (3) and \mathbf{Y}_1 is the scalar $\lambda_{\mathbf{T}}$ timing an identity matrix. Thus, left multiplying (17) by \mathbf{T} , taking the trace, and applying the power constraint in (3), we have

$$\lambda_{\mathbf{T}} = \text{tr}(\mathbf{T} \mathbf{Z} - \mathbf{T} \mathbf{X}_2 \mathbf{T}^* \mathbf{Y}_2) / q. \quad (19)$$

Note that (18) gives \mathbf{T} as a function of \mathbf{T} and $\lambda_{\mathbf{T}}$. Moreover, (19) gives $\lambda_{\mathbf{T}}$ as a function of \mathbf{T} . Thus, one way to get a \mathbf{T} is to iterate between (18) and (19) until convergence.

3.3. The iterative procedure

The GIA is as follows:

Step a. Set $k = 0$ and set the stopping threshold $\delta > 0$ and the maximum number of iterations k_{\max} .

Randomly initialize the precoders and relay processor for the 0th iteration: $\mathbf{F}_{1,0}$, $\mathbf{F}_{2,0}$, and \mathbf{T}_0 . Make sure they satisfy (2–3).

Step b. Calculate $\mathbf{C}_{ij,k} = \mathbf{H}'_i \gamma \mathbf{T}_k \mathbf{H}_j = \mathbf{H}'_i \mathbf{T}_k \mathbf{H}_j$ (since $\gamma = 1$). Also, use (6) to find $\Phi_{\mathbf{n}i,k}$ (replacing $\Phi_{\mathbf{n}i}$ and \mathbf{T} by $\Phi_{\mathbf{n}i,k}$ and \mathbf{T}_k , respectively).

Step c. Use (13–15) to find $\mathbf{F}_{j,k}$, $j = 1, 2$, (replacing \mathbf{F}_j , \mathbf{C}_{ij} and $\Phi_{\mathbf{n}i}$ by $\mathbf{F}_{j,k}$, $\mathbf{C}_{ij,k}$ and $\Phi_{\mathbf{n}i,k}$, respectively).

Step d. Use (17a–c) to find \mathbf{Z}_k , $\mathbf{X}_{1,k}$, and $\mathbf{Y}_{2,k}$ (replacing \mathbf{Z} , \mathbf{X}_1 , \mathbf{Y}_2 , \mathbf{F}_j , \mathbf{C}_{ij} , and $\Phi_{\mathbf{n}i}$ by \mathbf{Z}_k , $\mathbf{X}_{1,k}$, $\mathbf{Y}_{2,k}$, $\mathbf{F}_{j,k}$, $\mathbf{C}_{ij,k}$ and $\Phi_{\mathbf{n}i,k}$, respectively).

Step e. Use (19) to calculate $\lambda_{\mathbf{T},k}$ (replacing $\lambda_{\mathbf{T}}$, \mathbf{T} , \mathbf{Z} and \mathbf{Y}_2 by $\lambda_{\mathbf{T},k}$, \mathbf{T}_k , \mathbf{Z}_k , and $\mathbf{Y}_{2,k}$, respectively).

Step f. Use (17a) to calculate $\mathbf{Y}_{1,k}$ (replacing $\lambda_{\mathbf{T}}$ by $\lambda_{\mathbf{T},k}$).

Step g. Use (18) to calculate \mathbf{T}_{k+1} (replacing \mathbf{T} , \mathbf{Z} , \mathbf{X}_1 , \mathbf{Y}_1 and \mathbf{Y}_2 by \mathbf{T}_{k+1} , \mathbf{Z}_k , $\mathbf{X}_{1,k}$, $\mathbf{Y}_{1,k}$ and $\mathbf{Y}_{2,k}$, respectively).

Step h. If the precoder matrices converge ($dF_{j,k} = \|\mathbf{F}_{j,k} - \mathbf{F}_{j,k-1}\|_{\text{F}} < \delta$, for $j = 1, 2$), the relay processor converges ($d\mathbf{T}_k = \|\mathbf{T}_{k+1} - \mathbf{T}_k\|_{\text{F}} < \delta$), and the power constraint at the relay processor converges ($dq_k = |\text{tr}(\mathbf{Q}_k) - q| < \delta$), then set $\mathbf{F}_j = \mathbf{F}_{j,k}$ and $\mathbf{T} = \mathbf{T}_k$ and stop. Here, we do not have to check the power constraints at the end nodes since the closed form solution (14) always guarantees that they are satisfied.

Else if $k = k_{\max}$, set $\mathbf{F}_j = \mathbf{F}_{j,k}$ and $\mathbf{T} = \mathbf{T}_{k+1}$ and stop.

Otherwise, set $k = k + 1$ and return to step b.

After the above iteration is finished, remove any all zero columns from \mathbf{F}_1 and \mathbf{F}_2 . This does not change I_{E1} and I_{E2} in (10). However, it's needed so that (8) can be simplified to (10). Also, scale \mathbf{T} to satisfy the relay power constraint if the iteration terminated before convergence. Let $dp_{i,k} = |\sigma_{si}^2 \text{tr}\{\mathbf{F}_{i,k} \mathbf{F}_{i,k}^*\} - p_i|$. Observe that there is no need to check $dp_{1,k}$ and $dp_{2,k}$ in step h due to the use of the closed-form solutions for the precoders.

Below is the summary of the GIA approach. First, we transform the optimization problem in (11) into a root searching problem which attempts to solve for the relay processor \mathbf{T} using the system of nonlinear equations defined in (18). Note that, in (18) and the supplementary equations (17, 17a, 17b, 17c), the precoders \mathbf{F}_1 and \mathbf{F}_2 are no longer considered unknowns because they can be expressed in terms of \mathbf{T} using (14). Second, we solve (18) iteratively using step g. Thus, the convergence of GIA depends on the convergence of solving (18) using step g. As shown in [21], as long as the spectral radius (the largest magnitude of the eigenvalues) of the Jacobian matrix corresponding to (18) is less than 1 at the initial guess of the solution, an appropriate iterative approach will converge to one of the solutions. Thus, a proper selection of initial estimates will guarantee the convergence of

the proposed GIA. Extensive numerical studies have been performed and the convergent properties of the GIA are shown in Section 6.1. When the GIA converges, it will converge to a local extremum. It is difficult to prove whether the numerically derived solution is global optimum or not because the problem is highly nonlinear.

4. Decentralized design: protocol

As shown in (7), E_i ($i = 1, 2$) needs to know C_{ii} to remove the self interference. Actually, each of the nodes needs certain information such as C_{ij} , $i \neq j$, in order to design its processing matrix, decoder, etc. So, this section proposes a four-step protocol for the nodes to follow. To help explain why the protocol is constructed the way it is, perfect estimation for each channel sounding is assumed in this explanation.

The first step is for E_1 and E_2 to perform channel soundings so that the relay can estimate H_1 and H_2 . In the second step, the relay chooses a symmetric T —two possible ways are given in Section 5. Next, the relay picks γ so that (3) is satisfied. It can do this since it knows what precoders the end nodes will use (see step 4). In the third step, the relay performs two equivalent channel soundings, one precoded with γTH_1 and the other precoded with γTH_2 . From these two soundings, E_i ($i = 1, 2$) estimates the effective channel matrices C_{ii} and C_{ij} ($j \neq i$, $j = 1, 2$). The first term is what E_i needs to subtract its self-interference; C_{ii} is exactly what is left multiplying x_i in (5). The second term is the effective channel from E_j , through the relay, to itself. What is more, the transpose of the second term is the effective channel from itself to E_j —this is why the relay made T symmetric.

In the fourth step, E_i ($i = 1, 2$) designs its precoder F_i and decoder G_i without any signaling with the other end node. How can they do this and still have F_1 match G_2 and F_2 match G_1 ? Simple: E_1 uses a reproducible algorithm based on its estimate of C_{21} to design its precoder F_1 . E_2 follows the same algorithm with its estimate of C_{21} so that it can know E_1 's precoder F_1 . E_2 can thus design its decoder G_2 to match E_1 's precoder F_1 . E_2 and E_1 proceed in an analogous fashion so that E_2 's precoder F_2 and E_1 's decoder G_1 match as well. Recall that the relay already picked γ in the second step since it knew what precoders the end nodes would choose. Thus, one additional requirement for this reproducible algorithm is that it must be unaffected by a positive scaling of C_{21} . That is, the end node will result in the same precoder and decoder whether it is given C_{21} or ρC_{21} where ρ is any positive scalar. Though this last requirement may seem like a stringent restriction, the following example implementation of step 4 shows this is not necessarily the case. Moreover, the ability of the relay to calculate γ

before the end nodes design their precoders is very important as explained in Appendix 2.

To illustrate step 4, here is an example for F_1 and G_2 . The effective channel C_{21} is fixed since the relay has already picked T and γ . As in Section 3.1, let $r_1 = \min(d_1, \text{rank}(C_{21}))$ be the number of data streams E_1 will transmit. At E_1 , it first takes an SVD of C_{21}

$$C_{21} = U \Sigma V^*, \quad (20a)$$

where the singular values σ_1, σ_2 , etc., are in descending order, $U = [u_1 \dots u_{M_2}]$ and $v = v_1 \dots v_{M_1}$. Then, E_1 sets

$$F_1 = \alpha [v_1 e^{j\theta_1} \dots v_{r_1} e^{j\theta_{r_1}}], \quad (20b)$$

where the phase θ_l ($l = 1, \dots, r_1$) is chosen such that the first non-zero element of $v_l e^{j\theta_l}$ is positive. And, $\alpha > 0$ is chosen to satisfy the power constraint. E_2 also takes the same steps. E_2 takes a SVD of C_{21}

$$C_{21} = \tilde{U} \tilde{\Sigma} \tilde{V}^*, \quad (21a)$$

where $\tilde{U} = [\tilde{u}_1 \dots \tilde{u}_{M_2}]$ and $\tilde{V} = [\tilde{v}_1 \dots \tilde{v}_{M_1}]$. For each $l = 1, \dots, r_1$, it chooses ϕ_l such that the first non-zero element of $\tilde{v}_l e^{j\phi_l}$ is positive. Then, it chooses the scalar to satisfy the power constraint.

$$v_l e^{j\theta_l} = \tilde{v}_l e^{j\phi_l}, \quad \forall l. \quad (21b)$$

Thus, E_2 also gets F_1 . Equation (21b) depends on $\sigma_1, \dots, \sigma_{r_1}$ being distinct, something that can be assumed for physical systems. Now, E_2 proceeds to set

$$G_2 = \begin{bmatrix} \tilde{u}_1^* e^{-j\phi_1} \\ \vdots \\ \tilde{u}_{r_1}^* e^{-j\phi_{r_1}} \end{bmatrix}. \quad (22)$$

The end-to-end channel is thus

$$G_2 C_{21} F_1 = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{r_1} \end{bmatrix} \quad (23)$$

without any signaling between the two end nodes. Moreover, E_1 and E_2 would have gotten the same precoder and decoder, respectively, if they had been using ρC_{21} .

5. Decentralized design: relay processors

The decentralized protocol presented in Section 4 is very general and can support many possible relay processor designs as long as their T 's are symmetric. Two possible designs are presented here as examples.

5.1. Iterative symmetric design (ISA)

The sum mutual information, (10), depends on both the relay processor and the precoders. For this proposed decentralized design, the dependence on the precoders in (10) is removed to reduce the signaling loading as

discussed in Section 1. Furthermore, the noise propagated from the relay to the end nodes is ignored for convenience. That is, this design considers a simplified version of mutual information:

$$c_j(\mathbf{T}) = \left| \mathbf{I} + \rho^2 \mathbf{C}_{ij} \mathbf{C}_{ij}^* \right|, \quad i, j = 1, 2, i \neq j. \quad (24)$$

Here, $\rho^2 = p_i / \text{tr}(\Phi_{a_i})$ signifies the transmit SNR (the ratio between the total transmit signal power and the total receive thermal noise power) and c_j is related to the mutual information of the effective channel \mathbf{C}_{ij} . The problem is

$$\begin{aligned} \{\mathbf{T}_{\text{OPT}}\} &= \arg \max_{\mathbf{T}=\mathbf{T}', \text{tr}(\mathbf{T}\mathbf{T}^*)=1} (c_1(\mathbf{T}) + c_2(\mathbf{T})) \\ &= \arg \max_{\mathbf{T}=\mathbf{T}', \text{tr}(\mathbf{T}\mathbf{T}^*)=1} c_1(\mathbf{T}). \end{aligned} \quad (25)$$

The second “=” in (25) is due to

$$c_1(\mathbf{T}) = c_2(\mathbf{T}). \quad (26)$$

Equation (26) itself follows from $\mathbf{C}_{ij} = \mathbf{C}_{ji}'$ which, in turn, is a result of $\mathbf{T} = \mathbf{T}'$. The trace constraint is needed to keep the elements of \mathbf{T} from exploding to infinity. To search for such a \mathbf{T} , the augmented cost function

$$\zeta(\mathbf{T}) = c_1(\mathbf{T}) + \lambda \text{tr}(\mathbf{T}\mathbf{T}^*) \quad (27)$$

is introduced with the real Lagrange multiplier λ . Using the technique of variation, we obtain (see Appendix 3)

$$\text{tr} \left(\left(\overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 + \lambda \mathbf{T} \right) \mathbf{S} \right) = 0 \quad (28)$$

for every $M_R \times M_R$ symmetric matrix \mathbf{S} . In (28),

$$\mathbf{M}_1 \triangleq \rho^2 \mathbf{H}_1 \mathbf{H}_1^* \geq \mathbf{0}, \quad (28a)$$

$$\Gamma_a \triangleq |\Gamma| \Gamma^{-1} \text{ and } \Gamma \triangleq \mathbf{I} + \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 \mathbf{T}^* \overline{\mathbf{H}}_2 > \mathbf{0}. \quad (28b)$$

By Appendix 4, this in turn means that $\overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 + \lambda \mathbf{T}$ is skew-symmetric. That is,

$$-2\lambda \mathbf{T} = \overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 + \left(\overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 \right)'. \quad (29)$$

Noting that $\mathbf{T}^* = \overline{\mathbf{T}}$, right multiply (29) by $\overline{\mathbf{T}}$ and apply the trace constraint in (25) to get λ . Since $\text{tr}(\mathbf{N}\overline{\mathbf{T}}) = \text{tr} \left\{ \left(\mathbf{N}\overline{\mathbf{T}} \right)' \right\} = \text{tr}(\overline{\mathbf{T}}\mathbf{N}') = \text{tr}(\mathbf{N}'\overline{\mathbf{T}})$ with $\mathbf{N} \triangleq \overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1$, we have

$$\lambda = -\text{tr} \left(\overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 \overline{\mathbf{T}} \right). \quad (30)$$

Finally, plugging (30) into (29), we have

$$\mathbf{T} = \frac{\overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 + \left(\overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 \right)'}{2 \cdot \text{tr} \left(\overline{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 \overline{\mathbf{T}} \right)}. \quad (31)$$

Clearly, a \mathbf{T} satisfying (31) is a feasible solution to (25) as it is symmetric and satisfies the trace constraint of (25). As (31) gives an implicit expression of \mathbf{T} as a function of \mathbf{T} itself, this naturally leads to the following iterative procedure to get \mathbf{T} . Since (31) is a highly nonlinear equation of \mathbf{T} , averaging (step c) is used.

Step a Randomly initialize \mathbf{T}_0 as an $M_R \times M_R$ symmetric matrix satisfying the trace constraint of (25). Set $k = 0$. Also, set the stopping threshold δ and the maximum number of iterations k_{max} .

Step b Use (28b) to calculate $\Gamma_{a,k}$ (replacing Γ_a by $\Gamma_{a,k}$ and \mathbf{T} by \mathbf{T}_k).

Step c Set $\mathbf{T}_{k+1} = \beta \mathbf{T}_k + (1 - \beta) \hat{\mathbf{T}}_{k+1}$, where $0 \leq \beta < 1$ and $\hat{\mathbf{T}}_{k+1} = \frac{\overline{\mathbf{H}}_2 \Gamma_{a,k} \mathbf{H}'_2 \mathbf{T}_k \mathbf{M}_1 + \left(\overline{\mathbf{H}}_2 \Gamma_{a,k} \mathbf{H}'_2 \mathbf{T}_k \mathbf{M}_1 \right)'}{2 \cdot \text{tr} \left(\overline{\mathbf{H}}_2 \Gamma_{a,k} \mathbf{H}'_2 \mathbf{T}_k \mathbf{M}_1 \overline{\mathbf{T}}_k \right)}$. Note the similarity between $\hat{\mathbf{T}}_{k+1}$ and (31).

Also, set $d\eta_{k+1} = \left| \text{tr} \left\{ \mathbf{T}_{k+1} \mathbf{T}_{k+1}^* \right\} - 1 \right|$.

Step d. If $\frac{|c_1(\mathbf{T}_{k+1}) - c_1(\mathbf{T}_k)|}{c_1(\mathbf{T}_k)} < \delta$ and $d\eta_{k+1} < \delta$, set $\mathbf{T} = \mathbf{T}_{k+1}$ and stop.

Else if $k = k_{\text{max}}$, set $\mathbf{T} = \mathbf{T}_{k+1}$ and stop. Otherwise, set $k = k + 1$ and go to step b.

As in the GIA approach, we transform the optimization problem in (24) in the ISA approach into a root searching problem which attempts to solve for the relay processor \mathbf{T} for the system of nonlinear equations defined in (31). Thus, the convergence of ISA depends on the convergence of solving (31) iteratively using step c. As shown in [21], a proper selection of initial estimates will guarantee the convergence of the proposed ISA scheme. Extensive numerical studies have been performed and the convergent properties of the ISA are shown in Section 6.1. When the ISA converges, it will converge to a local extremum. Recall that (24) is an approximate mutual information formula to enable our decentralization. So, no optimality claim is made with regards to the actual sum mutual information.

5.2. RRANOMAX design

The second design is the relay processor design used in the RRANOMAX approach in [12]. It can be used when the relay noise covariance matrix is a scalar times an identity matrix, i.e., $\Phi_w = \sigma^2 \mathbf{I} > \mathbf{0}$. For completeness, we restate it below. Perform the SVD on $\mathbf{K} = 0.5[\mathbf{H}_2 \otimes \mathbf{H}_1, \mathbf{H}_1 \otimes \mathbf{H}_2]$:

$$\mathbf{K} = \mathbf{U}_K \boldsymbol{\Sigma}_K \mathbf{V}_K^* \quad (32)$$

Define $\mathbf{u}_{K,1}$ as the first column of \mathbf{U}_K . Then, the initial design of the relay processor is

$$\boldsymbol{\Omega} = \text{unvec}\{\bar{\mathbf{u}}_{K,1}\}. \quad (33)$$

The $M_R \times M_R$ symmetric matrix $\boldsymbol{\Omega}$ could be used as \mathbf{T} but it is not. The reason is due to the nature/distribution of its singular values. So, instead, perform the SVD on $\boldsymbol{\Omega}$:

$$\begin{aligned} \boldsymbol{\Omega} &= \mathbf{U}_\Omega \boldsymbol{\Sigma}_\Omega \mathbf{V}_\Omega^*, \text{ with } \boldsymbol{\Sigma}_\Omega \\ &= \text{DIAG}(\sigma_{\Omega_1}, \dots, \sigma_{\Omega_{M_R}}). \end{aligned} \quad (34)$$

The singular values in (34) are arranged in descending order. The \mathbf{T} will be obtained from $\boldsymbol{\Omega}$ by replacing $\sigma_{\Omega_1}, \dots, \sigma_{\Omega_{M_R}}$ by $\hat{\sigma}_{T_1}, \dots, \hat{\sigma}_{T_{M_R}}$. That is,

$$\begin{aligned} \mathbf{T} &= \mathbf{U}_T \hat{\boldsymbol{\Sigma}}_T \mathbf{V}_T^* \text{ with } \hat{\boldsymbol{\Sigma}}_T \\ &= \text{DIAG}(\hat{\sigma}_{T_1}, \hat{\sigma}_{T_2}, \dots, \hat{\sigma}_{T_{M_R}}). \end{aligned} \quad (35)$$

These new singular values are defined by

$$\hat{\sigma}_{T_k} = \begin{cases} \sqrt{\left(\mu - \frac{\sigma^2}{\lambda_k}\right)^+}, & k = 1, 2, \dots, L \\ 0, & \text{otherwise.} \end{cases} \quad (35a)$$

Here, $L = \min\{M_R, \min\{M_1, M_2\} + 1\}$ and μ is chosen so that $\hat{\sigma}_{T_1}^2 + \hat{\sigma}_{T_2}^2 + \dots + \hat{\sigma}_{T_{M_R}}^2 = 1$. In (35a),

$$\lambda_k = (\sigma_{1,k} + 1)(\sigma_{2,k} + 1), \quad (36)$$

where $\sigma_{i,k}$ is the k th singular value of \mathbf{H}_i (arranged in descending order).

6. Numerical results

Without loss of generality, assume that the noise covariance matrices $\boldsymbol{\Phi}_{ai}$ and $\boldsymbol{\Phi}_w$ are identity matrices. Also assume that the source covariance matrices are also identity matrices (i.e., $\sigma_{s1}^2 = \sigma_{s2}^2 = 1$). The numbers of antennas at the two end nodes are the same (i.e., $M_1 = M_2$) and are equal to 4. The number of antennas at the relay node M_R is either equal to M_1 or $2M_1$. Consider uncorrelated Rayleigh fading channels where all channel matrices are normalized such that the Frobenius norm of \mathbf{H}_j , $j = 1, 2$, is one. With the per-node power at each end node as $p_1 = p_2 = P$ and the relay power as $q = 2P$, we define ‘‘SNR’’ as $10 \log_{10}(P/M_1)$. The reason is that it is the dB value of the ratio between the total transmit power ($\text{tr}(\mathbf{F}_i \mathbf{F}_i^*)$) and the total thermal noise power ($\text{tr}(\boldsymbol{\Phi}_{ai})$) at an end node. Here, the channels and the noise propagated from the relay are not included in the ‘‘SNR’’ definition. The stop parameters for the ISA, δ , and k_{\max} are chosen to be 0.001 and 50, respectively. The stop parameters for

the GIA, δ , and k_{\max} are chosen to be 0.001 and 2000, respectively.

6.1. Convergence property of the GIA and ISA

Figure 2 shows the average number of iterations the GIA needed per ‘‘SNR’’ for 100 channel realizations. One set of points is for the $M_1 = M_2 = M_R = 4$ configuration while the other is for the $M_1 = M_2 = 4$ and $M_R = 8$ one. There have been other iterative transceiver designs very similar to the GIA (e.g., [22]). For those designs, it was common to see the number of iterations increasing with the SNR. It is interesting that the phenomenon is not seen here for both antenna configurations. Figure 3 shows an example convergence plot for the GIA for the $M_1 = M_2 = M_R = 4$ configuration and ‘‘SNR’’ = 0 dB. Observe that $dp_{1,k}$ and $dp_{2,k}$ are very small as expected due to the use of the closed-form solutions for the precoders.

Figure 4 shows the average number of iterations the ISA needed per ‘‘SNR’’ for 100 channel realizations. Again, the $M_1 = M_2 = M_R = 4$ and, $M_1 = M_2 = 4$ and $M_R = 8$ configurations are studied. It is very interesting that the trends for the two configurations are completely different. The $M_1 = M_2 = 4$ and $M_R = 8$ configuration sees a pretty constant average number of iterations needed over the different ‘‘SNR’’ values. On the other hand, the average number of iterations needed in the $M_1 = M_2 = M_R = 4$ configuration decreases as the ‘‘SNR’’ increases. Consequently, in the $M_1 = M_2 = M_R = 4$ configuration, there are many channel realizations for which the ISA converges almost right away. Figures 5 and 6 show example convergence plots for the ISA in both antenna configurations. For Figure 5 (the $M_1 = M_2 = M_R = 4$ configuration), note that the number of iterations for ‘‘SNR’’ = 0 dB is very large compared to that for ‘‘SNR’’ = 20 dB.

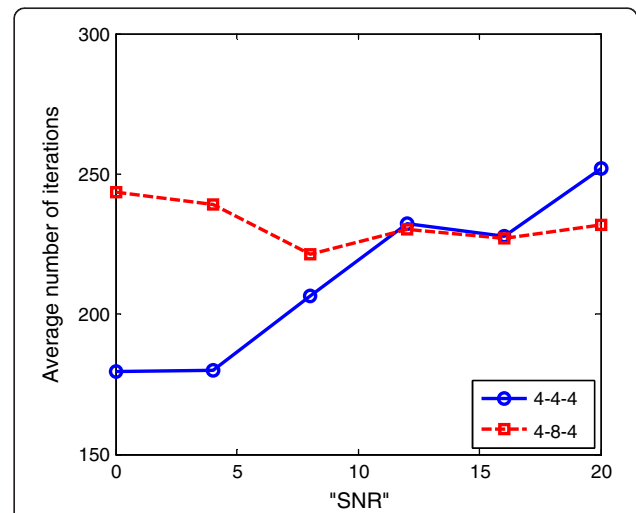
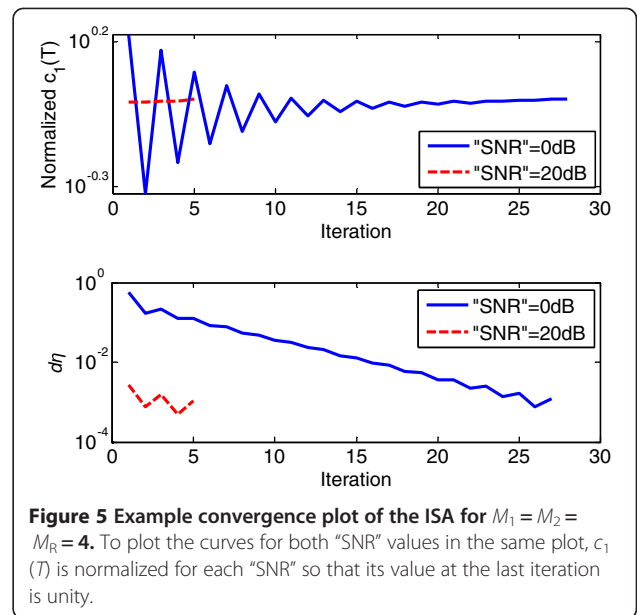
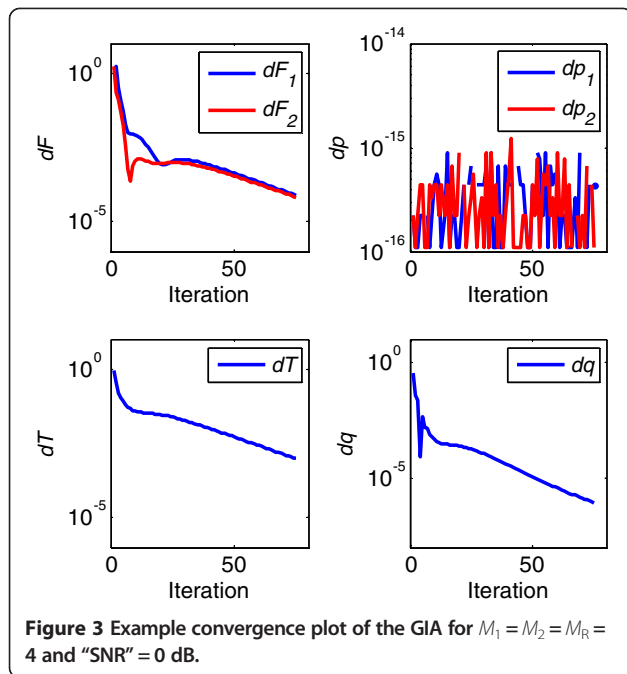


Figure 2 Average number of iterations the GIA needed per ‘‘SNR’’. In the legends, 4-4-4 denotes $M_1 = M_2 = M_R = 4$. Likewise, 4-8-4 denotes $M_1 = M_2 = 4$ and $M_R = 8$.

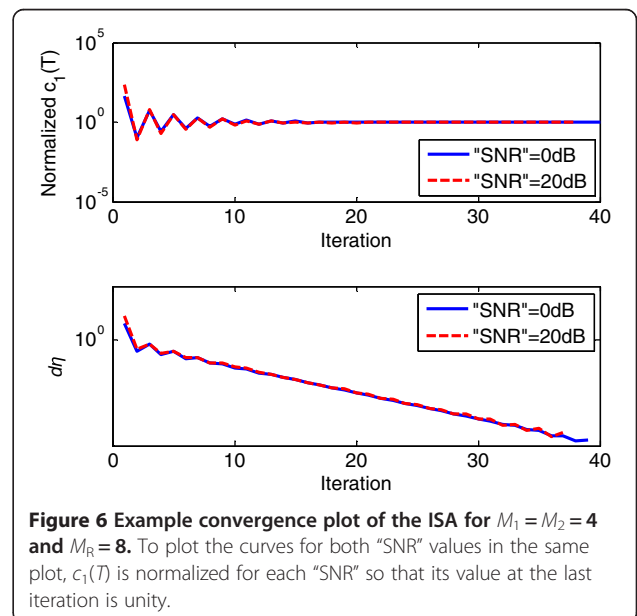
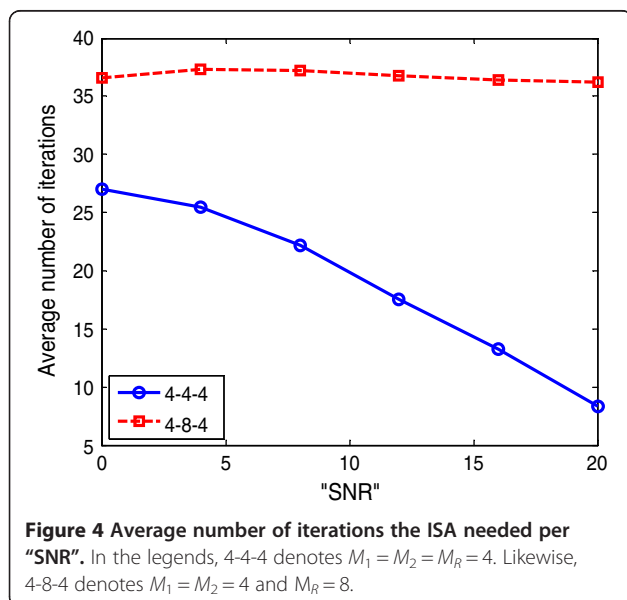


For Figure 6 (the $M_1 = M_2 = 4$ and $M_R = 8$ configuration), note that the number of iterations for "SNR" = 0 and 20 dB are basically the same. Both of these observations are consistent with our expectations after seeing Figure 4.

6.2. Sum mutual information performances

Recall that the decentralized signaling protocol proposed in Section 4 was flexible in terms of what relay processor \mathbf{T} the relay used. So here, we demonstrate the sum mutual information performance of the proposed protocol

with three relay designs. The first one is to choose $\mathbf{T} = \mathbf{I}$ and is denoted as "Identity". This is the simplest design. The second one is RRANOMAX where \mathbf{T} is given in (36). The third one is using the ISA (see Section 5.1). The protocol also allows for flexibility in the precoder and decoder design. For all three implementations of the protocol (one for each of the three relay processor designs), the SVD methodology described in (20a) to (22) is used for the precoder and decoder design. For reference, we also use (10) to evaluate the mutual information

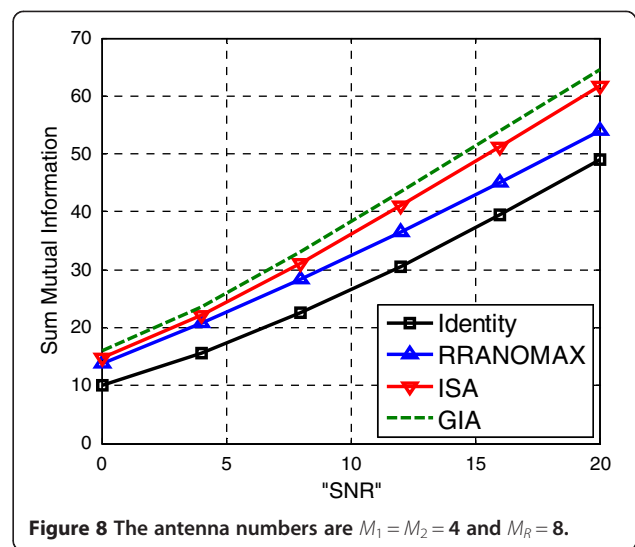
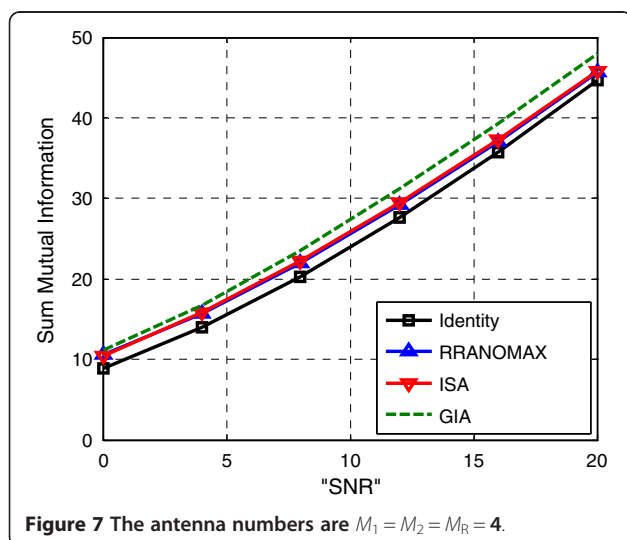


obtained using the proposed centralized GIA design. The GIA is described in Section 3.3.

Figures 7 and 8 show the sum mutual information, $I_{E1} + I_{E2}$, for the three practical (decentralized) designs and the centralized GIA design. The antennas numbers depend on the figure. $M_1 = M_2 = 4$ for both figures. $M_R = 4$ in Figure 7 and $M_R = 8$ in Figure 8. Each of the curves is obtained by averaging the results of 100 channel realizations. The number of data streams of each practical design at each “SNR” for each channel realization is chosen so that it provides the maximum sum mutual information. The number of data streams the GIA gives to E_i ($i = 1, 2$) is only determined in the last iteration; it is simply the number of non-zero columns of F_i .

From Figures 7 and 8, it is obvious that the “Identity” design of T has the worst performance and is greatly outperformed by the other three designs. The centralized “GIA” design has the best performance. The implementations of the proposed protocol with the “ISA” and “RRANOMAX” relay processor designs come in second and third place, respectively. By comparing Figure 7 with Figure 8, it is easy to see that the performance gaps increase as the number of relay antennas increases. In addition, the sum mutual information also becomes larger as the number of relay antennas increases. Both of these observed phenomena are probably due to the increased freedom at the relay—the size of T increases from 4×4 to 8×8 . Note that we normalize the Frobenius norms of all channel matrices to one.

Note that some kind of power loading at the precoders may increase the sum mutual information for the decentralized designs. However, performing the water filling procedure for power loading at the end nodes does *not* necessarily increase the mutual information. This is because the γ in (3) depends on the precoders.



When the power loading at the precoders changes the precoders $\{F_i\}$ according to the water filling procedure, γ has to change too (because γ depends on $\{F_i\}$). But then, the optimized power loading done previously for a different γ is no longer optimum in the sense of maximizing the sum mutual information for the new γ . Moreover, water filling requires the noise covariance matrices Φ_{ni} in (6) to be available at end nodes. But, Φ_{ni} is not available at the end node using the proposed decentralized protocol in Section 4. Thus, the water filling procedure is not applied here.

7. Conclusion

This article presents both practical and theoretical advances for MIMO analog network coding, a technique which requires only two time slots (excluding the time slots required for signaling) to complete a bi-directional communication between two end nodes. For the practical advance, the article proposes a decentralized joint transceiver and signaling design scheme which requires the system to work in a TDD mode and to have four channel soundings. With the proposed signaling scheme, each node gets the information needed for designing its own transmit and/or receive processors. The designs at all nodes are harmonized and coordinated such that no additional signaling overhead is needed. In particular, presented in this article is a novel iterative approach for designing a symmetric relay processor to maximize an approximate sum mutual information of the effective channels between the two end nodes. It is seen to converge quickly for all SNRs—highly desirable for a practical design.

For the theoretical advance, a novel iterative approach, named GIA, is proposed for jointly designing the precoders and decoders at the two end nodes and the

processor at the relay node. The goal of the design is to maximize the sum mutual information of the system subject to a per-node power constraint at each node. This is a centralized design and may not be practical, but can be used to generate benchmark results. The GIA alternately finds the precoders at the end nodes and the processor at the relay until all relevant parameters converge. In this article, the centralized GIA is employed to provide a performance benchmark for evaluating various implementations of the proposed decentralized design. It is remarkable that the performance of the proposed decentralized design is almost as good as the benchmark set by the centralized GIA. It is concluded that proposed decentralized scheme is high performing and can be a feasible way to implement analog network coding. Insights gained here may also possibly be used to enhance other existing designs in the analog network coding literature.

The proposed centralized GIA approach is very general and can be extended to deal with multiple relay nodes where the relay processors are determined sequentially. Moreover, it can be generalized to deal with arbitrary linear power constraints, including the practical per-antenna power constraint, where closed-form expressions for the precoders may not be available. For the proposed decentralized ISA design, an extension to multiple relays is much more difficult. As in the single-relay case, let each relay design its own relay processor. At this stage, it should calculate its power scaling parameter. However, it cannot; as it does not know the other channels and relay processors, it cannot figure out the SVD precoders and decoders of the end nodes. As the discussion in Appendix 2 still holds when there are multiple relays, having the end nodes calculate the power scaling parameters is not attractive. Having the relays communicate among themselves to determine their power scaling parameters will also involve a great deal of signaling as well. How to perform a practical decentralized design for the multiple relay case is thus a nontrivial problem for future research.

One challenging issue about the simultaneous multi-relay transmission in the analog network coding scheme for practical applications is the inevitable differences in propagation delays from the multiple relays to the end nodes. If the differences are large, the delay spread of the effective channel from multi-relay to an end node will be also large. Then it will require a large guard time. Moreover, if the relays are not synchronized well, the effective delay spread will vary in different transmissions. This will make it very challenging for each end node to cancel its own signal and to detect the desired signal. Thus, the simultaneous multi-relay transmission mechanism is not practical for the analog network coding scheme if all relays cannot be well synchronized. In that case, a sequential multi-relay transmission mechanism can

be implemented and the multi-relay system is reduced into multiple single-relay systems.

Endnotes

^aIt was also presented as a part of Enoch Lu's Ph.D. thesis [16].

^bIt was also presented in IEEE Globecom 2011 [11] and in Enoch Lu's Ph.D. thesis [16].

Appendix 1

The technique of variation can be used to obtain a stationary point of a scalar function $\xi(\mathbf{T})$ in (16) with respect to the matrix variable \mathbf{T} :

$$\nabla \xi(\mathbf{T}) = 0 \quad (37)$$

Let $\mathbf{T}(\varepsilon) = \mathbf{T} + \varepsilon \Delta_{\mathbf{T}}$, where ε is a scalar perturbation parameter and $\Delta_{\mathbf{T}}$ is an arbitrary matrix with the same dimension as \mathbf{T} . Instead of working with (37), one can also find a stationary point by finding a \mathbf{T} which satisfies

$$\left. \frac{\partial \xi(\mathbf{T} + \varepsilon \Delta_{\mathbf{T}})}{\partial \varepsilon} \right|_{\varepsilon=0} = 0 \quad (38)$$

for every matrix $\Delta_{\mathbf{T}}$. This is the route the technique of variation takes. Through laborious but straightforward manipulations of (38), we get

$$\begin{aligned} 0 = & \lambda_{\mathbf{T}} \text{tr} \left\{ \Delta_{\mathbf{T}} \left(\sum_{n=1}^2 \sigma_{s_k}^2 \mathbf{H}_k \mathbf{F}_k \mathbf{F}_k^* \mathbf{H}_k^* + \Phi_{\mathbf{w}} \right) \mathbf{T}^* \right. \\ & \left. + \mathbf{T} \left(\sum_{n=1}^2 \sigma_{s_k}^2 \mathbf{H}_k \mathbf{F}_k \mathbf{F}_k^* \mathbf{H}_k^* + \Phi_{\mathbf{w}} \right) \Delta_{\mathbf{T}}^* \right\} \\ & + \frac{1}{\ln 2} \sum_{i=1}^2 \text{tr} \left\{ \left(\left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} - \Phi_{\mathbf{n}i}^{-1} \right) \right. \\ & \quad \left. \times \mathbf{H}'_i (\Delta_{\mathbf{T}} \Phi_{\mathbf{w}} \mathbf{T}^* + \mathbf{T} \Phi_{\mathbf{w}} \Delta_{\mathbf{T}}^*) \bar{\mathbf{H}}_i \right\} \\ & + \frac{1}{\ln 2} \sum_{i=1}^2 \text{tr} \left\{ \sigma_{s_j}^2 \left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} \right. \\ & \quad \left. \times \left(\mathbf{H}'_i \Delta_{\mathbf{T}} \mathbf{H}_j \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{H}'_i \Delta_{\mathbf{T}}^* \bar{\mathbf{H}}_i \right) \right\}. \end{aligned} \quad (39)$$

Since $\Delta_{\mathbf{T}}$ is arbitrary, we can replace $\Delta_{\mathbf{T}}$ in (39) by $\Delta_{\mathbf{T}}$ times the imaginary unit. Doing this yields

$$\begin{aligned}
 0 = & \lambda_{\mathbf{T}} \text{tr} \left\{ \Delta_{\mathbf{T}} \left(\sum_{n=1}^2 \sigma_{s_k}^2 \mathbf{H}_k \mathbf{F}_k \mathbf{F}_k^* \mathbf{H}_k^* + \Phi_{\mathbf{w}} \right) \mathbf{T}^* \right. \\
 & \left. - \mathbf{T} \left(\sum_{n=1}^2 \sigma_{s_k}^2 \mathbf{H}_k \mathbf{F}_k \mathbf{F}_k^* \mathbf{H}_k^* + \Phi_{\mathbf{w}} \right) \Delta_{\mathbf{T}}^* \right\} \\
 & + \frac{1}{\ln 2} \sum_{i=1}^2 \text{tr} \left\{ \left(\left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} - \Phi_{\mathbf{n}i}^{-1} \right) \right. \\
 & \quad \times \mathbf{H}'_i \left(\Delta_{\mathbf{T}} \Phi_{\mathbf{w}} \mathbf{T}^* - \mathbf{T} \Phi_{\mathbf{w}} \Delta_{\mathbf{T}}^* \right) \bar{\mathbf{H}}_i \left. \right\} \\
 & + \frac{1}{\ln 2} \sum_{i=1}^2 \text{tr} \left\{ \sigma_{s_j}^2 \left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} \right. \\
 & \quad \times \left(\mathbf{H}'_i \Delta_{\mathbf{T}} \mathbf{H}_j \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* - \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{H}_j \Delta_{\mathbf{T}}^* \bar{\mathbf{H}}_i \right) \left. \right\}. \tag{40}
 \end{aligned}$$

Removing the terms containing $\Delta_{\mathbf{T}}^*$ by summing (39) and (40), and after some manipulations (mainly cycling the matrices inside the trace operators so that $\Delta_{\mathbf{T}}$ is in the left side of each term), we have

$$\begin{aligned}
 0 = & \text{tr} \left\{ \Delta_{\mathbf{T}} \left[\lambda_{\mathbf{T}} \left(\sum_{n=1}^2 \sigma_{s_k}^2 \mathbf{H}_k \mathbf{F}_k \mathbf{F}_k^* \mathbf{H}_k^* + \Phi_{\mathbf{w}} \right) \mathbf{T}^* \right. \right. \\
 & \left. + \frac{1}{\ln 2} \sum_{i=1}^2 \Phi_{\mathbf{w}} \mathbf{T}^* \bar{\mathbf{H}}_i \left(\left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} - \Phi_{\mathbf{n}i}^{-1} \right) \mathbf{H}'_i \right. \\
 & \left. \left. + \frac{1}{\ln 2} \sum_{i=1}^2 \sigma_{s_j}^2 \mathbf{H}_j \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* \left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} \mathbf{H}'_i \right] \right\}. \tag{41}
 \end{aligned}$$

As $\Delta_{\mathbf{T}}$ is arbitrary, $\Delta_{\mathbf{T}}$ can be the conjugate of the term inside the square brackets in (41). We thus conclude that the term inside the square brackets must be zero because $\text{tr}(\mathbf{A}\mathbf{A}^*) = 0$ implies $\mathbf{A} = 0$. Thus, we have the following expression which will lead to (17, 17a–c):

$$\begin{aligned}
 & \left(\sum_{n=1}^2 \sigma_{s_k}^2 \mathbf{H}_k \mathbf{F}_k \mathbf{F}_k^* \mathbf{H}_k^* + \Phi_{\mathbf{w}} \right) \mathbf{T}^* \lambda_{\mathbf{T}} \\
 & + \frac{1}{\ln 2} \Phi_{\mathbf{w}} \mathbf{T}^* \sum_{i=1}^2 \left(\mathbf{H}'_i \right)^* \left(\left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} - \Phi_{\mathbf{n}i}^{-1} \right) \mathbf{H}'_i \\
 & = - \frac{1}{\ln 2} \sum_{i=1}^2 \sigma_{s_j}^2 \mathbf{H}_j \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* \left(\sigma_{s_j}^2 \mathbf{C}_{ij} \mathbf{F}_j \mathbf{F}_j^* \mathbf{C}_{ij}^* + \Phi_{\mathbf{n}i} \right)^{-1} \mathbf{H}'_i. \tag{42}
 \end{aligned}$$

Appendix 2

Let us assume that the relay is unable to calculate γ before the end nodes design their precoders. Certainly, this

means that the relay is unable to know what precoders the end nodes will choose without some signaling from the end nodes—if it knew the precoders, it could simply solve (3) and get γ . Consequently, γ or the information needed to calculate γ must be signaled to the relay.

Consider the first strategy of having γ signaled to the relay. Necessarily, an end node has to calculate γ then. The current protocol however does not give either end node enough information though. Take E_1 for example. Assume E_1 knows all the information the protocol in Section 4 gives it: $\mathbf{H}_1' \mathbf{T} \mathbf{H}_2$, $\mathbf{H}_1' \mathbf{T} \mathbf{H}_1$, \mathbf{F}_1 , and \mathbf{F}_2 (note $\gamma \mathbf{H}_1' \mathbf{T} \mathbf{H}_2$ and $\gamma \mathbf{H}_1' \mathbf{T} \mathbf{H}_1$ are changed to $\mathbf{H}_1' \mathbf{T} \mathbf{H}_2$ and $\mathbf{H}_1' \mathbf{T} \mathbf{H}_1$ as the relay does not know γ). Even with all of this information, E_1 cannot compute *even* one of the three terms of \mathbf{Q} in (3): $\mathbf{T} \Phi_{\mathbf{w}} \mathbf{T}^*$, $\sigma_{s_1}^2 \mathbf{T} \mathbf{H}_1 \mathbf{F}_1 \mathbf{F}_1^* \mathbf{H}_1^* \mathbf{T}^*$, and $\sigma_{s_2}^2 \mathbf{T} \mathbf{H}_2 \mathbf{F}_2 \mathbf{F}_2^* \mathbf{H}_2^* \mathbf{T}^*$. Regarding the first term of \mathbf{Q} , it does not even know \mathbf{T} and $\Phi_{\mathbf{w}}$. Regarding the other two terms of \mathbf{Q} , it does not know $\mathbf{T} \mathbf{H}_1$ and $\mathbf{T} \mathbf{H}_2$. Clearly, considerable signaling is needed for an end node to be able to calculate γ .

Now, consider the alternative strategy of having the end nodes signal whatever is needed by the relay so that it can calculate γ . For example, let each end node perform an equivalent channel sounding with its precoder. The relay thus knows $\mathbf{H}_1 \mathbf{F}_1$ and $\mathbf{H}_2 \mathbf{F}_2$. Combined with the information the relay already has, it can now calculate γ . No matter which strategy, not enabling the relay to calculate γ before the end nodes design their precoders necessitates adding signaling to the protocol. Or, to put it positively, choosing the precoder design in step 4 so that the relay can calculate γ in step 2 helps reduce the amount of signaling.

Appendix 3

As in Appendix 1, the technique of variation is employed here. Note though that the *symmetric* matrix constraint here will lead to us a slightly different development. First, replace the \mathbf{T} in (27) by $\mathbf{T}(\varepsilon) = \mathbf{T} + \varepsilon \Delta$. Here, ε is a real scalar and Δ is an arbitrary *symmetric* matrix. Second, evaluate the derivative of $\zeta(\mathbf{T} + \varepsilon \Delta)$ with respect to ε at $\varepsilon = 0$ and set it equal to 0. Through laborious but straightforward manipulations, we have

$$\begin{aligned}
 & \text{tr} \left(\left(\bar{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 \mathbf{T} \mathbf{M}_1 + \lambda \mathbf{T} \right) \Delta^* \right) \\
 & + \text{tr} \left(\Delta \left(\mathbf{M}_1 \mathbf{T}^* \bar{\mathbf{H}}_2 \Gamma_a \mathbf{H}'_2 + \lambda \mathbf{T}^* \right) \right) \\
 & = 0 \tag{43}
 \end{aligned}$$

where \mathbf{M}_1 and Γ_a are defined in (28a) and (28b), respectively. For any symmetric Δ , Δ times the imaginary unit is also symmetric. We can thus replace Δ in (43) by Δ times the imaginary unit. Doing this yields

$$\begin{aligned}
 & -\text{tr}\left(\left(\bar{\mathbf{H}}_2\Gamma_a\mathbf{H}'_2\mathbf{T}\mathbf{M}_1 + \lambda\mathbf{T}\right)\Delta^*\right) \\
 & + \text{tr}\left(\Delta\left(\mathbf{M}_1\mathbf{T}^*\bar{\mathbf{H}}_2\Gamma_a\mathbf{H}'_2 + \lambda\mathbf{T}^*\right)\right) \\
 & = 0.
 \end{aligned} \tag{44}$$

Removing the terms containing Δ by summing (43) and (44) we have

$$\text{tr}\left(\left(\bar{\mathbf{H}}_2\Gamma_a\mathbf{H}'_2\mathbf{T}\mathbf{M}_1 + \lambda\mathbf{T}\right)\Delta^*\right) = 0 \tag{45}$$

For convenience, replace the notation of the symmetric matrix Δ^* by \mathbf{S} , which leads to (28).

Appendix 4

For notational convenience, let \mathcal{B} denote the set of all $N \times N$ symmetric complex matrices. This appendix will prove the following lemma.

Lemma: Matrix \mathbf{L} is $N \times N$. $\text{tr}\{\mathbf{L}\mathbf{S}\} = 0$ for all $\mathbf{S} \in \mathcal{B}$ if and only if \mathbf{L} is skew-symmetric.

Proof of reverse direction: By definition, $\mathbf{L} = -\mathbf{L}'$. Applying this, it can be shown for any $\mathbf{S} \in \mathcal{B}$ that $\text{tr}\{\mathbf{L}\mathbf{S}\} = \text{tr}\{\mathbf{S}\mathbf{L}'\} = -\text{tr}\{\mathbf{S}\mathbf{L}\} = -\text{tr}\{\mathbf{L}\mathbf{S}\}$. This in turn implies that $\text{tr}\{\mathbf{L}\mathbf{S}\} = 0$ as desired.

Proof of forward direction: One can always write $\mathbf{L} = \mathbf{L}_A + \mathbf{L}_B$ where $\mathbf{L}_A = \mathbf{L}/2 + \mathbf{L}'/2$ is symmetric and $\mathbf{L}_B = \mathbf{L}/2 - \mathbf{L}'/2$ is skew-symmetric. Since \mathbf{L}_A is symmetric, $\bar{\mathbf{L}}_A = \mathbf{L}_A^*$. Choosing $\mathbf{S} = \bar{\mathbf{L}}_A$, the hypothesis implies that $0 = \text{tr}\{\mathbf{L}\bar{\mathbf{L}}_A\} = \text{tr}\{\mathbf{L}_A\bar{\mathbf{L}}_A\} + \text{tr}\{\mathbf{L}_B\bar{\mathbf{L}}_A\}$. The second term, $\text{tr}\{\mathbf{L}_B\bar{\mathbf{L}}_A\}$, is 0 because \mathbf{L}_B is skew-symmetric (see proof of reverse direction). So, the first term must be zero, i.e., $0 = \text{tr}\{\mathbf{L}_A\bar{\mathbf{L}}_A\}$. Clearly, this last equality means $\mathbf{L}_A = \mathbf{0}$, making $\mathbf{L} = \mathbf{L}_B$ skew-symmetric.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to express their gratitude to Prof. Dante Youla for his comments.

Received: 29 July 2012 Accepted: 16 February 2013

Published: 10 April 2013

References

1. P Popovski, H Yomo, Wireless network coding by amplify-and-forward for bi-directional traffic flows. *IEEE Commun. Lett.* **11**(1), 16–18 (2007)
2. S Abdallah, IN Psaromiligkos, Blind channel estimation for amplify-and-forward two-way relay networks employing M-PSK modulation. (arXiv, 2012), <http://arxiv.org/abs/1101.4207>. Accessed 11 March 2013
3. Y Jia, A Vosoughi, Impact of channel estimation error upon sum-rate in amplify-and-forward two-way relaying systems, in *Proceedings of the 11th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2010)* pp. 20–23. June 2010
4. F Gao, R Zhang, Y-C Liang, Optimal channel estimation and training design for two-way relay networks. *IEEE Trans. Commun.* **57**(10), 3024–3033 (2009)
5. N Lee, H Park, J Chun, Linear precoder and decoder design for two-way AF MIMO relaying system, in *Proceedings of the IEEE 67th Vehicular Technology Conference (VTC)* pp. 11–14. May 2008

6. X Tang, Y Hua, Optimal design of non-regenerative MIMO wireless relays. *IEEE Trans. Wirel. Commun.* **6**(4), 1398–1407 (2007)
7. K-J Lee, KW Lee, H Sung, I Lee, Sum-rate maximization for two-way MIMO amplify-and-forward relaying systems, in *Proceedings of the IEEE 69th Vehicular Technology Conference (VTC)* pp. 26–29. April 2009
8. T Taniguchi, Y Karasawa, N Nakajima, Design of two way multiantenna AF relay systems under perfect CSI, in *Proceedings of the 4th European Conference on Antennas and Propagation (EuCAP)* 12–16 April 2010
9. E Lu, J Li, I-T Lu, Novel MMSE design for joint MIMO processing in analog network coding schemes, in *Proceedings of the 4th International Conference on Signal Processing and Communication Systems (ICSPCS)* 13–15 Dec. 2010
10. J Li, E Lu, I-T Lu, Joint MMSE designs for analog network coding and different MIMO relaying schemes: a unified approach and performance benchmarks (Phys. Commun, 2012). doi:10.1016/j.phycom.2012.02.003
11. E Lu, Z You, I-T Lu, Max capacity design for two-way MIMO relay systems, in *Proceedings of the IEEE Global Telecommunications Conference (Globecom)* 5–9 Dec. 2011
12. F Roemer, M Haardt, A low-complexity relay transmit strategy for two-way relaying with MIMO amplify and forward relays, in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* 14–19 March 2010
13. EA Jorswieck, A Sezgin, Transmit strategies for the MIMO two-way amplify-forward channel with multiple relays and MMSE receiver, in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* 14–19 March 2010
14. AY Panah, RW Heath Jr, Sum-rate of MIMO two-way relaying with imperfect CSI, in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* 14–19 March 2010
15. F Roemer, M Haardt, Tensor-based channel estimation (TENICE) for two-way relaying with multiple antennas and spatial reuse, in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* 19–24 April 2009
16. E Lu, *Practical and theoretical advances in MIMO systems*. Ph.D. thesis, Polytechnic Institute of NYU, USA, 2012
17. DM Pozar, *Microwave Engineering*, 3rd edn. (Wiley, New York, 2005)
18. T Sarkar, M Salazar-Palma, E Mokole, *Physics of Multiantenna Systems and Broadband Processing* (Wiley, New York, 2008)
19. D Guo, S Shamai, S Verdú, Mutual information and minimum mean-square error in Gaussian channels. *IEEE Trans. Inf. Theory* **51**(4), 1261–1282 (2005)
20. A Scaglione, P Stoica, S Barbarossa, GB Giannakis, H Sampath, Optimal designs for space-time linear precoders and decoders. *IEEE Trans. Signal Process.* **50**, 1051–1064 (2002)
21. A Ostrowski, *Solution of Equations and Systems of Equations* (Academic Press, New York, 1966) (see Theorem 22.1 in page 162)
22. I-T Lu, J Li, E Lu, Novel MMSE precoder and decoder designs subject to per-antenna power constraint for uplink multiuser MIMO systems, in *Proceedings of the 3rd International Conference on Signal Processing and Communication Systems (ICSPCS)*, 28–30 Sept. 2009

doi:10.1186/1687-6180-2013-71

Cite this article as: Lu et al.: Decentralized practical design and centralized benchmark for analog network coding. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:71.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com