

Maximum Likelihood Channel Estimation in Decode-and-Forward Relay Networks

Feifei Gao

Institute for Infocomm Research, A*STAR
21 Heng Mui Keng Terrace
119613, Singapore
Email: feifeigao@ieee.org

Tao Cui

Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, USA
Email: taocui@caltech.edu

Arumugam Nallanathan

Division of Engineering
King's College London
London, United Kingdom
Email: nallanathan@ieee.org

Abstract—In this paper, we provide a complete study on the training based channel estimation for relay networks that employ the decode-and-forward (DF) scheme. Since multiple relay nodes are geographically distributed over the service region, channel estimation is different from the traditional way in that each relay has its own individual power constraint. We consider the maximum likelihood (ML) channel estimation and derive closed form solutions for the optimal training as well as for the optimal power allocation. It is seen that the optimal power allocation follows a multi-level waterfilling structure.

I. INTRODUCTION

Employing multiple antennas can boost the system capacity by transmitting multiple data streams [1] and enhance the transmission reliability using space-time coding (STC) techniques [2]. Unfortunately, packing more than one antennas onto a small mobile terminal faces many difficulties such as the size limitation and the hardware complexity. In order to overcome this limitation, one would refer to the relay networks, where spacial diversity is achieved when relays are deemed as “virtual antennas” for the desired user [3]- [5].

It has been pointed out in [6] that the channel estimation and optimal training design for *amplify-and-forward* (AF) relay networks is quite different from that in the traditional point-to-point systems, which motivates a new look into the channel estimation for DF relay networks. For DF relay networks, overall transmission is divided into two phases. Since the relays decode during Phase I and re-encode the information bits during Phase II, Phase I and Phase II are actually separated. Hence, the main scheme of channel estimation is similar to that in the traditional point-to-point systems. However, since relays are geographically distributed and different relays may come from different types of mobile terminals, the individual power constraint for each relay has to be considered. These individual power constraints form the major challenge and most times bring difficulties to find closed form solutions during the optimization process. Although there exist many training based channel estimation methods for traditional point-to-point systems [7]- [9], optimal channel estimation with individual power constraint for each antenna has not yet been considered either in relay networks or in the traditional multi-input multi-output (MIMO) systems, to the best of the authors' knowledge.

In DF relay networks, nevertheless, a total power constraint can also be included when there exists a central control unit (CCU). Although CCU in this case cannot allot power to each relay from a common power pool, it can still determine how

much the summation of the power is within each relay's own power constraints (to keep the budget of the desired user).

In this work, we provide a complete study for ML based channel estimation. The training design includes designing the training sequence and determining the power of each relay within its own power constraints. We show that the optimal power distribution has a multi-level waterfilling type structure while the corresponding training sequence can be obtained from efficient algorithms.

II. SYSTEM MODEL OF DF RELAY NETWORKS

Consider a wireless network with M randomly placed relay nodes \mathbb{R}_i , $i = 1, \dots, M$, one source node \mathbb{S} , one destination node \mathbb{D} , and M_I interfering nodes \mathbb{I}_j , $j = 1, \dots, M_I$ operating in the same frequency band. Every node has only a single antenna that cannot transmit and receive simultaneously. The channel between any two nodes is assumed quasi-stationary Rayleigh flat fading in that it is constant within one frame but may vary from frame to frame. Denote the channel from \mathbb{S} to \mathbb{R}_i as g_i , from \mathbb{R}_i to \mathbb{D} as h_i , from \mathbb{I}_j to \mathbb{R}_i as f_{ji} , from \mathbb{I}_j to \mathbb{D} as q_j respectively; namely $g_i \in \mathcal{CN}(0, \sigma_{g_i})$, $h_i \in \mathcal{CN}(0, \sigma_{h_i})$, $f_{ji} \in \mathcal{CN}(0, \sigma_{f_{ji}})$ and $q_j \in \mathcal{CN}(0, \sigma_{q_j})$. Note that, the interference, if any, affects both the relays and the destination, which brings an undesired scenario. We assume perfect synchronization among \mathbb{S} , \mathbb{R}_i and \mathbb{D} . However, no synchronization assumption is made for interfering nodes.

The training is accomplished by the following two phases, each containing N consecutive time slots. For Phase I, the transmitter broadcasts the signal \mathbf{s} to relays and the destination. The received signals at \mathbb{R}_i is expressed as

$$\mathbf{r}_i = g_i \mathbf{s} + \sum_{j=1}^{M_I} f_{ji} \mathbf{s}_{j1} + \mathbf{n}_{ri} \quad (1)$$

where \mathbf{s}_{j1} is the equivalent based-band signal from the j th interference during Phase I, and \mathbf{n}_{ri} is the white complex Gaussian noise at the i th relay. During Phase II, the i th relay sends out¹ \mathbf{s}_i of length N and the destination receives

$$\mathbf{y} = \underbrace{[\mathbf{s}_1 \ \mathbf{s}_2 \ \dots \ \mathbf{s}_M]}_{\mathbf{C}} \underbrace{\begin{bmatrix} h_1 \\ \vdots \\ h_M \end{bmatrix}}_{\mathbf{h}} + \underbrace{\sum_{j=1}^{M_I} q_j \mathbf{s}_{j2}}_{\mathbf{n}_d} + \mathbf{n}_d \quad (2)$$

¹In DF relay networks, relays during the second phase will send out new training symbols to estimate h_i 's at the destination only.

where \mathbf{s}_{j2} is the signal from the j th interference during Phase II, and $\mathbf{n}_{d2} \in \mathcal{CN}(0, N_0 \mathbf{I})$ represents the complex white Gaussian noise vector at \mathbb{D} . The equivalent colored noise \mathbf{n}_d has the covariance

$$\mathbf{R}_n = \mathbb{E}\{\mathbf{n}_d \mathbf{n}_d^H\} = N_0 \mathbf{I} + \mathbb{E}\left\{\left(\sum_{j=1}^{M_I} q_j \mathbf{s}_{j2}\right)\left(\sum_{j=1}^{M_I} q_j \mathbf{s}_{j2}\right)^H\right\} \quad (3)$$

which is assumed known to the destination.

The task of the channel estimation includes estimating both g_i and h_i . The estimation of g_i can be done exactly as in the traditional single-input single-output (SISO) system and the corresponding discussion will be omitted. In the remaining of the paper, we will only focus on estimating h_i . Meanwhile, $N \geq M$ is required since there are M unknown channels to be estimated. Assume, during the training process, each relay can maximally provide a power of p_i . Then the individual power constraint of the i th relay could be expressed as

$$[\mathbf{C}^H \mathbf{C}]_{ii} \leq p_i. \quad (4)$$

To offer a more general discussion at this point, we assume that there exists a CCU, and the overall training power consumed from relays is limited by P ; namely

$$\text{tr}(\mathbf{C}^H \mathbf{C}) \leq P. \quad (5)$$

Note that CCU in distributed relay network cannot allocate power to each relay from a common power pool but rather control the power level of each relay within its own power constraint. Clearly, P should be less than $\sum_{i=1}^M p_i$, otherwise the total power constraint is redundant. Meanwhile, P should also be greater than $\min_i p_i$, otherwise all the individual constraints are redundant. In the following, we assume that $\min_i p_i < P < \sum_{i=1}^M p_i$.

III. MAXIMUM LIKELIHOOD CHANNEL ESTIMATION

A. Problem Formulation

The ML estimation of \mathbf{h} is obtained as

$$\hat{\mathbf{h}}_{ML} = (\mathbf{R}_n^{-\frac{1}{2}} \mathbf{C})^\dagger \mathbf{R}_n^{-\frac{1}{2}} \mathbf{y} = (\mathbf{C}^H \mathbf{R}_n^{-1} \mathbf{C})^{-1} \mathbf{C}^H \mathbf{R}_n^{-1} \mathbf{y} \quad (6)$$

where $(\cdot)^\dagger$ denotes the pseudo inverse and the error covariance matrix is

$$\mathbb{E}\{(\hat{\mathbf{h}}_{ML} - \mathbf{h})^H (\hat{\mathbf{h}}_{ML} - \mathbf{h})\} = (\mathbf{C}^H \mathbf{R}_n^{-1} \mathbf{C})^{-1}. \quad (7)$$

The optimal \mathbf{C} can be found by solving the following constrained optimization problem:

$$\begin{aligned} \text{(P1)} \quad & \min_{\mathbf{C}} \quad \text{tr}((\mathbf{C}^H \mathbf{R}_n^{-1} \mathbf{C})^{-1}) \\ & \text{s.t.} \quad [\mathbf{C}^H \mathbf{C}]_{ii} \leq p_i, \quad i = 1, \dots, M \\ & \quad \text{tr}(\mathbf{C}^H \mathbf{C}) \leq P. \end{aligned} \quad (8)$$

Without loss of generality, we assume p_i are arranged in non-decreasing order and define $\mathbf{p} = [p_1, p_2, \dots, p_M]^T$. We first note that P1 is equivalent to the following problem:

$$\begin{aligned} \text{(P2)} \quad & \min_{\mathbf{D}} \quad \text{tr}((\mathbf{D}^H \mathbf{R}_n^{-1} \mathbf{D})^{-1}) \\ & \text{s.t.} \quad \mathbf{D}^H \mathbf{D} \text{ is diagonal} \\ & \quad \mathbf{d}(\mathbf{D}^H \mathbf{D}) \succ^w \mathbf{p}, \quad \text{tr}(\mathbf{D}^H \mathbf{D}) \leq P \end{aligned} \quad (9)$$

where \succ^w is notation for the weak majorization defined in [10]. The equivalency can be shown by referring to a similar procedure in [10] with some slight modification, which is omitted for brevity.

Let $\mathbf{d}(\mathbf{A})$ and $\boldsymbol{\lambda}(\mathbf{A})$ denote the vectors formed by the diagonal elements and the eigen-values of \mathbf{A} , both arranged in non-decreasing order. Then, the optimal \mathbf{C} can be found from \mathbf{D} based on the following steps:

Algorithm 1: Finding \mathbf{C} of P1 from \mathbf{D} of P2

- 1) Find diagonal elements of $\mathbf{C}^H \mathbf{C}$ such that $\mathbf{d}(\mathbf{C}^H \mathbf{C}) \leq \mathbf{p}$ and $\mathbf{d}(\mathbf{C}^H \mathbf{C}) \prec \boldsymbol{\lambda}(\mathbf{D}^H \mathbf{D})$, where \prec denotes the majorization operation [10].
- 2) Find \mathbf{U}_C such that $\mathbf{U}_C \mathbf{D}^H \mathbf{D} \mathbf{U}_C^H$ has the diagonal elements $\mathbf{d}(\mathbf{C}^H \mathbf{C})$.
- 3) Construct $\mathbf{C} = \mathbf{D} \mathbf{U}_C^H$.

The algorithm for the second step was shown in [11, Sect. IV-A] and the one for the first step will be provided later.

Now it is still unclear how to handle the optimization in P2. Without loss of generality, we can represent \mathbf{D} as $\mathbf{Q} \boldsymbol{\Sigma}_D^{\frac{1}{2}}$ where \mathbf{Q} is an $N \times M$ orthonormal matrix and $\boldsymbol{\Sigma}_D^{\frac{1}{2}}$ is a real diagonal matrix with diagonal element $\sigma_{D,i}^{\frac{1}{2}} \geq 0$. Since the column order of \mathbf{Q} can be changed arbitrarily with the corresponding interchanging of $\sigma_{D,i}^{\frac{1}{2}}$, we can assume that $\sigma_{D,i}^{\frac{1}{2}}$ are arranged in non-decreasing order. The optimization problem then becomes

$$\begin{aligned} \min_{\mathbf{Q}, \sigma_{D,i}} \quad & \text{tr}((\boldsymbol{\Sigma}_D^{\frac{1}{2}} \mathbf{Q}^H \mathbf{R}_n^{-1} \mathbf{Q} \boldsymbol{\Sigma}_D^{\frac{1}{2}})^{-1}) \\ \text{s.t.} \quad & \mathbf{Q}^H \mathbf{Q} = \mathbf{I} \\ & \sum_{i=1}^k \sigma_{D,i} \leq \sum_{i=1}^k p_i, \quad k = 1, \dots, M \\ & \sigma_{D,i} \leq \sigma_{D,i+1}, \quad \sigma_{D,i} \geq 0, \quad \sum_{i=1}^M \sigma_{D,i} \leq P. \end{aligned} \quad (10)$$

Suppose the eigen-value decomposition (EVD) of \mathbf{R}_n is $\mathbf{R}_n = \mathbf{U}_n \boldsymbol{\Sigma}_n \mathbf{U}_n^H$, where \mathbf{U}_n is an $N \times N$ unitary matrix and $\boldsymbol{\Sigma}_n = \text{diag}\{\sigma_{n,1}, \dots, \sigma_{n,N}\}$ is a diagonal matrix. Since the column order of \mathbf{U}_n can be changed arbitrarily if the diagonal elements in $\boldsymbol{\Sigma}_n$ are interchanged accordingly, we can always assume that $\sigma_{n,i}$ are arranged in non-decreasing order. We get to the following lemma:

Lemma 1: The optimal \mathbf{Q} to (10) is $\mathbf{U}_n [\mathbf{I}_M, \mathbf{0}_{M, N-M}^T]^T$ and the optimal $\sigma_{D,i}$ can be found from

$$\begin{aligned} \min_{\sigma_{D,i}} \quad & \sum_{i=1}^M \frac{\sigma_{n,i}}{\sigma_{D,i}} \\ \text{s.t.} \quad & \sum_{i=1}^k \sigma_{D,i} \leq \sum_{i=1}^k p_i, \quad k = 1, \dots, M \\ & \sigma_{D,i} \leq \sigma_{D,i+1}, \quad \sigma_{D,i} \geq 0, \quad \sum_{i=1}^M \sigma_{D,i} \leq P. \end{aligned} \quad (11)$$

Proof: We first prove that the optimal $\boldsymbol{\Sigma}_D^{\frac{1}{2}} \mathbf{Q}^H \mathbf{R}_n^{-1} \mathbf{Q} \boldsymbol{\Sigma}_D^{\frac{1}{2}}$ must be a diagonal matrix. Note that the optimization can be

separately conducted for \mathbf{Q} and $\sigma_{D,i}$. The objective function can be equivalent written as

$$\text{tr}((\Sigma_D^{\frac{1}{2}} \mathbf{Q}^H \mathbf{R}_n^{-1} \mathbf{Q} \Sigma_D^{\frac{1}{2}})^{-1}) = \text{tr}((\mathbf{Q}^H \mathbf{R}_n^{-1} \mathbf{Q})^{-1} \Sigma_D^{-1}). \quad (12)$$

Suppose the eigenvalues of $\mathbf{Q}^H \mathbf{R}_n^{-1} \mathbf{Q}$ are $\lambda_i, i = 1, \dots, M$ which are arranged in non-decreasing order. From [12] we know

$$\text{tr}((\mathbf{Q}^H \mathbf{R}_n^{-1} \mathbf{Q})^{-1} \Sigma_D^{-1}) \geq \sum_{i=1}^M \frac{1}{\lambda_{M-i+1} \sigma_{D,i}} \quad (13)$$

where the equality hold when the eigen-matrix of $\mathbf{Q}^H \mathbf{R}_n^{-1} \mathbf{Q}$ is an appropriate permutation matrix. From [13, Theorem 10, pp. 209], we know $\lambda_i \leq \frac{1}{\sigma_{n,M-i+1}}$. Therefore, (13) is lower bounded by $\sum_{i=1}^M \frac{\sigma_{n,i}}{\sigma_{D,i}}$. Note that this lower bound can be achieved when setting $\mathbf{Q} = \mathbf{U}_n [\mathbf{I}_M, \mathbf{0}_{M,N-M}^T]^T$ in (10), which is in turn, the optimal value of (10). This shows that, the optimal training should apply all energy on the eigen-modes that correspond to the smallest interference levels, i.e., the smallest $\sigma_{n,i}$, which agrees with our intuition very well. ■

B. Waterfilling Solution to (11)

We can remove the constraints $\sigma_{D,i} \geq 0$ and $\sigma_{D,i} \leq \sigma_{D,i+1}$ since an optimal solution always satisfies them. This point will also be clear later.

Since $p_1 < P < \sum_{i=1}^M p_i$, there must exist an integer $k^* \in \{1, 2, \dots, M-1\}$, such that $\sum_{i=1}^{k^*} p_i < P$ while $\sum_{i=1}^{k^*+1} p_i \geq P$. Therefore, the constraints $\sum_{i=1}^k \sigma_{D,i} \leq \sum_{i=1}^k p_i$, for $k = k^* + 1, \dots, M$ are redundant.

The Lagrangian of the optimization problem is written as

$$\mathcal{L} = \sum_{i=1}^M \frac{\sigma_{n,i}}{\sigma_{D,i}} + \sum_{k=1}^{k^*} \mu_k \left(\sum_{i=1}^k \sigma_{D,i} - \sum_{i=1}^k p_i \right) + \nu \left(\sum_{i=1}^M \sigma_{D,i} - P \right) \quad (14)$$

where μ_k and ν are Lagrange multipliers. The solution can be found from Karush-Kuhn-Tucker (KKT) optimization conditions:

$$\begin{aligned} -\frac{\sigma_{n,k}}{\sigma_{D,k}^2} + \sum_{i=k}^{k^*} \mu_i + \nu &= 0, \quad 1 \leq k \leq k^* \\ -\frac{\sigma_{n,k}}{\sigma_{D,k}^2} + \nu &= 0, \quad k^* + 1 \leq k \leq M \\ \mu_k \left(\sum_{i=1}^k \sigma_{D,i} - \sum_{i=1}^k p_i \right) &= 0, \quad 1 \leq k \leq k^* \\ \nu \left(\sum_{i=1}^M \sigma_{D,i} - P \right) &= 0, \quad \mu_k \geq 0, \quad \nu \geq 0. \end{aligned}$$

First of all, $\sum_{i=1}^M \sigma_{D,i} = P$ must hold at the optimal point. Otherwise, $\nu = 0$ and $-\frac{\sigma_{n,k}}{\sigma_{D,k}^2} + \nu = 0$ cannot hold. Without loss of generality, we suppose at the optimal point, only m out of k^* μ_k 's are non-zero (the equality of the corresponding constraint holds) and denote these m μ_k 's as $\mu_{k_i}, i = 1, \dots, m$

with $k_1 < k_2 < \dots < k_m$. The assumption indicates $\mu_k = 0$ for $1 \leq k < k_1$. Then,

$$-\frac{\sigma_{n,k}}{\sigma_{D,k}^2} + \sum_{i=1}^m \mu_{k_i} + \nu = 0, \quad 1 \leq k \leq k_1 \quad (15)$$

$$\sum_{i=1}^{k_1} \sigma_{D,i} = \sum_{i=1}^{k_1} p_i. \quad (16)$$

Define $\nu_1 = \sum_{i=1}^m \mu_{k_i} + \nu$. We have $\sigma_{D,k} = \sqrt{\frac{\sigma_{n,k}}{\nu_1}}$ for $1 \leq k \leq k_1$. This is exactly the weighted waterfilling by considering $\sqrt{1/\nu_1}$ as the water level and $\sqrt{\sigma_{n,k}}$ as weight for patches $1 \leq k \leq k_1$ with zero bottom levels. Due to these zero bottom levels, the water level $\sqrt{1/\nu_1}$ can be explicitly calculated as $\frac{\sum_{i=1}^{k_1} p_i}{\sum_{i=1}^{k_1} \sqrt{\sigma_{n,i}}}$. Since $\sigma_{n,i}$'s are arranged in non-decreasing order, we can directly see that $\sigma_{D,k_1} \geq \dots \geq \sigma_{D,1}$. We then go on considering $\mu_{k_2}, \mu_{k_3}, \dots, \mu_{k_m}$ sequentially. The general equations are written here:

$$-\frac{\sigma_{n,k}}{\sigma_{D,k}^2} + \sum_{i=j}^m \mu_{k_i} + \nu = 0, \quad k_{j-1} < k \leq k_j, \quad 2 \leq j \leq m \quad (17)$$

$$\sum_{i=1}^{k_j} \sigma_{D,i} = \sum_{i=1}^{k_j} p_i. \quad (18)$$

Equation (18) is in fact equivalent to

$$\sum_{i=k_{j-1}+1}^{k_j} \sigma_{D,i} = \sum_{i=k_{j-1}+1}^{k_j} p_i. \quad (19)$$

Define $\nu_j = \sum_{i=j}^m \mu_{k_i} + \nu$. There is $\sigma_{D,k} = \sqrt{\frac{\sigma_{n,k}}{\nu_j}}$ for $k_{j-1} < k \leq k_j$ and the corresponding water level is $\sqrt{1/\nu_j}$. For the same reason, $\sigma_{D,k}$ is in non-decreasing order for $k_{j-1} < k \leq k_j$. Moreover, since $\nu_j = \nu_{j-1} - \mu_{k_{j-1}} \leq \nu_{j-1}$, the water level $\sqrt{1/\nu_j}$ is also arranged in non-decreasing order. Combining the fact that $\sqrt{\sigma_{n,i}}$ is arranged in non-decreasing order, we know that $\sigma_{D,k_{j-1}+1} \geq \sigma_{D,k_{j-1}}$ and therefore, $\sigma_{D,i}, i = 1, \dots, k_m$ should be in non-decreasing order. Meanwhile, the water level $\sqrt{1/\nu_j}$ can be explicitly calculated as $\frac{\sum_{i=k_{j-1}+1}^{k_j} p_i}{\sum_{i=k_{j-1}+1}^{k_j} \sqrt{\sigma_{n,i}}}$.

Lastly, we have

$$-\frac{\sigma_{n,k}}{\sigma_{D,k}^2} + \nu = 0, \quad k_m < k \leq M, \quad (20)$$

$$\sum_{i=k_m+1}^M \sigma_{D,i} = P - \sum_{i=1}^{k_m} p_i. \quad (21)$$

The corresponding water level is $\sqrt{1/\nu} = \frac{P - \sum_{i=1}^{k_m} p_i}{\sum_{i=k_m+1}^M \sqrt{\sigma_{n,i}}}$, and $\sigma_{D,k} = \sqrt{\frac{\sigma_{n,k}}{\nu}}$ for $k_m < k \leq M$. Similarly, $\sigma_{D,k}$ should be in non-decreasing order for $k_m < k \leq M$ and $\sigma_{D,k_m+1} \geq \sigma_{D,k_m}$.

The above discussion not only provides the insight on how to design the optimizing algorithm but also confirms the validity of omitting the constraints $\sigma_{D,i} \leq \sigma_{D,i+1}$ and $\sigma_{D,i} \geq 0$ in the first place. The solution structure follows a

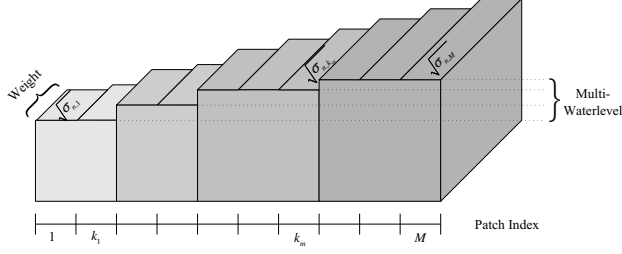


Fig. 1. Illustration on weighted multi-level waterfilling.

weighted multi-level waterfilling with multiple water levels at $\{\sqrt{1/\nu_j}, \sqrt{1/\nu}\}$ and the weight for the i th patch is $\sqrt{\sigma_{n,i}}$, as shown in Fig. 1. The cutting point $k_j, j = 1, \dots, m$ can be obtained from the testing, which follows a similar approach as that in the traditional waterfilling problem.

Algorithm 2: Multi-Level Waterfilling

- 1) Set $j = 0, k_0 = 0$.
- 2) For each $k_j + 1 \leq t \leq k^*$, calculate the water level $\frac{\sum_{i=k_j+1}^t p_i}{\sum_{i=k_j+1}^t \sqrt{\sigma_{n,i}}}$ (assuming patches $k_j + 1$ to t are saturated) and the water level $\frac{P - \sum_{i=1}^{k_j} p_i}{\sum_{i=k_j+1}^M \sqrt{\sigma_{n,i}}}$ (assuming patch $k_j + 1$ to M are saturated). If the water level $\frac{P - \sum_{i=1}^{k_j} p_i}{\sum_{i=k_j+1}^M \sqrt{\sigma_{n,i}}}$ is the lowest, go to 3). Otherwise, if index t_0 gives the lowest water level, set $k_{j+1} = t_0$ and calculate $\sigma_{D,k} = \frac{\sqrt{\sigma_{n,k}} \sum_{i=k_j+1}^{k_{j+1}} p_i}{\sum_{i=k_j+1}^{k_{j+1}} \sqrt{\sigma_{n,i}}}$ for $k_j + 1 \leq k \leq k_{j+1}$. If $k_{j+1} = k^*$, then $j := j + 1$ and go to 3); otherwise, $j := j + 1$ and go back to 2).
- 3) Calculate $\sigma_{D,k} = \frac{\sqrt{\sigma_{n,k}} (P - \sum_{i=1}^{k_j} p_i)}{\sum_{i=k_j+1}^M \sqrt{\sigma_{n,i}}}$, for $k_j + 1 \leq k \leq M$.

Under the worst case, the water level need to be calculated $\frac{k^*(k^*+3)}{2}$ times.

C. Optimal Solution to The Original Problem (8)

After getting the optimal $\sigma_{D,i}$, we need to construct the original \mathbf{C} for problem P1, following Algorithm 1. We here provide a simple way to realize the first step in Algorithm 1. Denote $\mathbf{d}(\mathbf{C}^H \mathbf{C}) = [c_1, c_2, \dots, c_M]$.

Algorithm 3: Finding $\mathbf{d}(\mathbf{C}^H \mathbf{C})$

- 1) Set $c_i = \sigma_{D,i}$ for all i .
- 2) From $i = M : -1 : 2$, if $c_i > p_i$, then set $c_i := p_i$ and set $c_{i-1} := c_{i-1} + (c_i - p_i)$.

The validity of Algorithm 3 is proved as follows:

Proof:

1) $\mathbf{d}(\mathbf{C}^H \mathbf{C}) \prec \mathbf{d}(\Sigma_D)$: From the initialization, we know $\sum_{i=1}^k c_i \geq \sum_{i=1}^k \sigma_{D,i}$ and $\sum_{i=1}^M c_i = \sum_{i=1}^M \sigma_{D,i}$. From the algorithm, the excessive part $(c_k - p_k)$ will be included into c_{k-1} . This does not change the equality $\sum_{i=1}^M c_i = \sum_{i=1}^M \sigma_{D,i}$. Meanwhile, since more value are included into c_{k-1} , the inequality $\sum_{i=1}^k c_i \geq \sum_{i=1}^k \sigma_{D,i}$ for $k = 1, \dots, M-1$ will be kept.

2) $c_1 \leq c_2 \leq \dots \leq c_M$ and $\mathbf{d}(\mathbf{C}^H \mathbf{C}) \leq \mathbf{p}$: From the algorithm, we know $c_k \leq p_k$ and c_k is already in non-decreasing order after the initialization. If at the current step, c_k is smaller than p_k , c_{k-1} will be kept unchanged and $c_{k-1} \leq c_k$ still holds (remember c_k won't be decreased in all previous steps). If on the other side c_k is greater than or equal to p_k , then $c_k := p_k$ and $c_{k-1} := c_{k-1} + (c_k - p_k)$. However, at next step, this c_{k-1} will be upperbounded by p_{k-1} and the excessive part $c_{k-1} - p_{k-1}$ will be added to c_{k-2} . Bearing in mind that p_k 's are arranged in non-decreasing order, we know that $c_{k-1} \leq c_k$ still holds. This process continues until $k = 2$.

The speciality happens for c_1 since there is no behavior regarding to whether c_1 is greater or less than p_1 . Therefore, we only need to prove that $c_1 \leq p_1$ and $c_1 \leq c_2$. These two things can be proved together. If $c_2 \leq p_2$ still holds after getting the increment, then, there will be no increment for c_1 . In this case, the final c_1 is $\sigma_{D,1}$ and the proof is completed. Otherwise, $c_2 > p_2$ and the excessive part will be added to c_1 . We can find a maximal integer $r_0 \in \{2, 3, \dots, M\}$, such that c_k is equal to p_k for $2 \leq k \leq r_0$ when the algorithm finishes. Then, the final c_1 is $\sigma_{D,1} + \sum_{i=2}^{r_0} (\sigma_{D,i} - p_i)$. From the optimization process, we know

$$\sum_{i=1}^{r_0} p_i \geq \sum_{i=1}^{r_0} \sigma_{D,i}. \quad (22)$$

Then

$$p_1 \geq \lambda_{D,1} + \sum_{i=2}^{r_0} (\sigma_{D,i} - p_i) = c_1 \quad (23)$$

can be derived. Since the final value of c_2 is p_2 in this case, we arrive at $c_1 \leq c_2$. ■

After obtaining $\mathbf{d}(\mathbf{C}^H \mathbf{C})$, we can construct \mathbf{C} from the remaining two steps in Algorithm 1.

IV. SIMULATION RESULTS

In this section, we numerically examine the performance of our proposed channel estimation algorithms as well as the optimal training designs, under various scenarios. The signal-to-noise-ratio is defined as $\text{SNR} = P/MN/N_0 = P/MN$ (average power over time and spacial index).

The channels h_i 's are assumed as circularly symmetric complex Gaussian random variables with variances $\sigma_{h,i}$, unified according to $\sum_{i=1}^M \sigma_{h,i} = M$. The channel covariance matrices \mathbf{R}_h have the following structures:

$$[\mathbf{R}_h]_{i,j} = \sqrt{\sigma_{h,i} \sigma_{h,j}} \varepsilon_1^{|i-j|},$$

where $\varepsilon_1 < 1$ is a real scalar that affects the correlation factors among channels. Interference covariance matrices \mathbf{R}_n in our example has the similar structure as \mathbf{R}_h , where a real scalar $\varepsilon_2 < 1$ is used to control the correlation factors among noise. The average interference power is assumed to be 10 times of the noise so that $\text{tr}(\mathbf{R}_n)/M = 11N_0$.

The training sequence $\hat{\mathbf{s}}_i$ that is the scalar multiple of the optimal \mathbf{s}_i will be named as the optimal training sequence (optimal T). Correspondingly, the L2 norm of the optimal \mathbf{s}_i will be referred to as the optimal power allocation (optimal P).

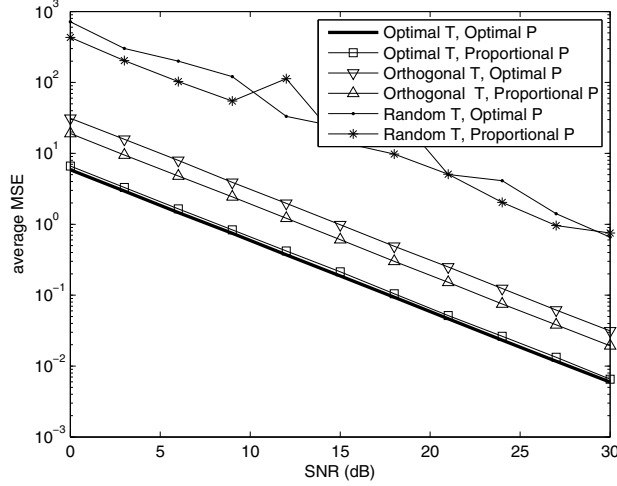


Fig. 2. Comparison between different training and power allocation for ML based channel estimation, with $\varepsilon_1 = 0.9$, $\varepsilon_2 = 0.9$, $M = N = 4$.

The proportional power allocation (proportional P) is defined as $\hat{p}_i = \frac{P_i}{\sum_{i=1}^M P_i} P$.

The comparison is mainly conducted between the optimal training sequence with both the orthogonal training (orthogonal T) and the random training (random T). Therefore, the following 6 different types of the training scenarios will be examined: “Optimal T, Optimal P”, “Optimal T, Proportional P”, “Orthogonal T, Optimal P”, “Orthogonal T, Proportional P”, “Random T, Optimal P”, “Random T, Proportional P”. For all numerical examples, we take 10000 Monte-Carlo runs for average.

To exhibit the effect of the correlated channel and the colored interference, we here adopt a relatively large ε_1 and ε_2 as $\varepsilon_1 = \varepsilon_2 = 0.9$. In Fig. 2, we display the MSEs of ML channel estimation versus SNR of for 6 different training scenarios under the system parameters $M = N = 4$. We see that, the optimal training with the optimal power is slightly better than the optimal training with the proportional power. The orthogonal training under both power allocations have more than 6 dB SNR loss than the optimal one. The performance of the random training has around 20 dB SNR loss compared to the optimal one and is not stable since we assume the smallest valid N . This phenomenon has also been observed in the channel estimation for AF relay networks [6]. We then increase the N to 8 while keeping all other parameters unchanged and show different MSEs in Fig. 3. Most observations are the same as those in Fig. 2 except that the performance of the random training become more stable and is better than that of the orthogonal training. Another observation is that, the optimal power allocation derived may not necessarily be the best type of the power allocation, if combined with orthogonal training and the random training.

V. CONCLUSIONS

In this paper, we study the training based channel estimation for DF based relay networks. The major challenge here is the

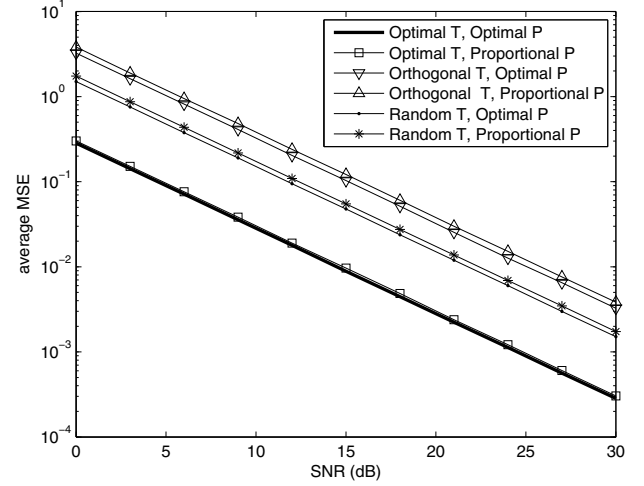


Fig. 3. Comparison between different training and power allocation for ML based channel estimation, with $\varepsilon_1 = 0.9$, $\varepsilon_2 = 0.9$, $M = N/2 = 4$.

individual constraint of each relay node. To provide a thorough study, we also include a CCU which brings the total power constraint. The popular ML approach is considered and we find a multi-level waterfilling solution. Numerical examples have been provided from which we find that the optimal training and the optimal power allocation are both important to achieve the best channel estimation.

REFERENCES

- [1] I. E. Telatar, “Capacity of multi-antenna Gaussian channels,” *Eur. Trans. Telecom.*, vol. 10, pp. 585-595, Nov. 1999.
- [2] V. Tarokh, N. Seshadri, and A. R. Calderbank, “Space time codes for high data rate wireless communication: performance criterion and code construction,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 744-765, 1998.
- [3] J. N. Laneman and G. W. Wornell, “Distributed space time block coded protocols for exploiting cooperative diversity in wireless networks,” *IEEE Trans. Inform. Theory*, vol. 49, pp. 2415-2425, Oct. 2003.
- [4] S. Yiu, R. Schober, and L. Lampe, “Distributed space time block coding,” *IEEE Trans. Commun.*, vol. 54, pp. 1195-1206, July, 2006.
- [5] Y. Jing and B. Hassibi, “Distributed space time coding in wireless relay networks,” *IEEE Trans. Wireless Commun.*, vol. 5, pp. 3524-3536, Dec. 2006.
- [6] F. Gao, T. Cui and A. Nallanathan, “On channel estimation and optimal training design for amplify and forward relay networks”, to appear in *IEEE Trans. Wireless Commun.*, vol. 7, May 2008.
- [7] I. Barhumi, G. Leus, and M. Moonen, “Optimal training design for MIMO OFDM systems in mobile wireless channels,” *IEEE Trans. Signal Processing*, vol. 51, pp. 1615-1624, June 2003.
- [8] M. Biguesh and A. B. Gershman, “Training based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals,” *IEEE Trans. Signal Processing*, vol. 54, pp. 884-893, Mar. 2006.
- [9] Y. Liu, T. F. Wong, and W. W. Hager, “Training signal design for estimation of correlated MIMO channels with colored interference,” *IEEE Trans. Signal Processing*, vol. 55, pp. 1486-1497, Apr. 2007.
- [10] D. P. Palomar, M. A. Lagunas, and J. M. Cioffi, “Optimum linear joint transmit-receive processing for MIMO channels with QoS constraints,” *IEEE Trans. Signal Processing*, vol. 52, pp. 1179-1197, May 2004.
- [11] P. Viswanath and V. Anantharam, “Optimal sequences and sum capacity of synchronous CDMA systems,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 1984-1991, Sept. 1999.
- [12] F. Zhang and Q. Zhang, “Eigenvalue inequalities for matrix product,” *IEEE Trans. Autom. Control*, vol. 51, pp. 1506-1509, Sept. 2006.
- [13] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, New York: Wiley, 1999.