EUSIPCO 2013 1569747771

DECENTRALIZED DIRECTION FINDING USING PARTLY CALIBRATED ARRAYS

Wassim Suleiman, Marius Pesavento, Abdelhak Zoubir

Institute of Telecommunications, Technische Universität Darmstadt
(suleiman, zoubir)@spg.tu-darmstadt.de, pesavento@nt.tu-darmstadt.de

ABSTRACT

We consider direction-finding in partly calibrated arrays composed of multiple identically oriented subarrays. The subarrays are assumed to possess a shift invariance structure that can be exploited for search free direction of arrival (DoA) estimation. We propose a fully distributed DoA estimation scheme that is based on the averaging consensus algorithm, in which the subarrays communicate only locally with their neighboring subarrays to exchange local averages of their measurements and received signal to iteratively compute global estimates in the network. The proposed scheme eliminates communication bottlenecks and the need for a centralized computation center. Our algorithm is based on subspace methods which is originally devised for DoA estimation in centralized systems. We show that in our fully distributed DoA estimation scheme, the number of jointly estimated directions can be larger than the number identifiable by each individual subarray.

Index Terms— Averaging consensus algorithm, subarray systems, ESPRIT, distributed DoA estimation.

1. INTRODUCTION

DoA estimation of narrowband sources has attracted significant attention [1], since it is applied in many array processing applications including sonar, radar, and seismic exploration. Subspace methods such as MUSIC [2], MODE [3] and WSF [4] are computationally efficient direction finding methods that exhibit the super resolution DoA estimation property. However, they can only be applied in fully calibrated sensor arrays where the locations of all sensors are known. Other subspace methods such as ESPRIT [5], RARE [6], Multiple Invariant ESPRIT [7] and Multiple Invariant MUSIC and MODE [8] can be used in partly calibrated arrays composed of perfectly calibrated subarrays with unknown subarray displacements. Our algorithm is based on the ESPRIT algorithm which exploits the specific shift invariance structure of the array to efficiently compute the parameters of interest in a search-free procedure.

The original algorithms mentioned above are centralized, and require centralized processing of the measurements recorded from all subarrays. Hence each subarray needs to forward its measurements data to a computation center in which the DoA algorithm is implemented. Due to transmit power limitation, pathloss and interference the subarrays far from the computation center will not be able to transfer their measurements on a direct link to the computation center and generally multi-hop communication may have to be established to forward all the measurements to the computation center. Thus, subarrays will forward their measurements to adjacent subarrays which are located closer to the computation center. This requires the application of efficient routing protocols and increases the communication load on the subarrays located in the vicinity of the computation center which renders them the communication bottlenecks. Therefore such centralized multi-hop networks generally do not scale well as adding new subarrays to the system may result in a complete system breakdown. In this case, new communication routes between the new subarrays and the computation center needs to be found, which adds to the overall complexity. Furthermore, maintaining the routes is very difficult, for example in case of sensor failure. These well-established drawbacks of multi-hop communication motivate the use of decentralized subspace estimation based on patch processing as in [9], or decentralized subspace tracking as in [10] and [11]. In [9] and [10] the authors show how their algorithms can be used for DoA estimation in calibrated arrays. However according to our best knowledge, decentralized DoA estimation in partly calibrated arrays has not been considered yet.

This paper is organised as follows. In Section 2, the problem of DoA estimation is formulated and the signal model is presented. Section 3 briefly describes the conventional ESPRIT algorithm. In Section 4, the decentralized power method [9] for distributed estimation of the signal subspace is revised. Based on this we propose an averaging consensus scheme referred to as decentralized ESPRIT (d-ESPRIT) for DoA estimation in subarray sensor networks. Section 5 describes the averaging protocol used in the d-ESPRIT algorithm. Finally, we will verify our algorithm by simulation.

We use lowercase and uppercase bold letters to denote vectors and matrices, respectively. The transpose, the Hermitian transpose and the complex conjugation are denoted by $(.)^T$, $(.)^H$ and $(.)^*$, respectively. The Kronecker product is denoted by $\otimes$.

2. SIGNAL MODEL

We consider an arbitrary array composed of p subarrays. The displacements between these subarrays assumed to be unknown. For simplicity we will restrict our presentation to arrays composed of identical and identically oriented uniform linear subarrays [1] composed of m antennas separated by distance d. Consider that $q \leq p(m-1)$ signals from stationary narrowband far-field sources impinge the array from directions $\theta_i, i = 1, \ldots, q$. Let $\boldsymbol{x}_k(t)$ denotes the measurements vector of the kth subarray at time instant $t \in \mathbb{N}$, then [5]

$$\boldsymbol{x}_k(t) = \boldsymbol{A}_k \boldsymbol{s}(t) + \boldsymbol{n}_k(t), \tag{1}$$

where $\boldsymbol{s}(t) \in \mathbb{C}^q$ is the vector of impinging signals, and $\boldsymbol{A}_k \in \mathbb{C}^{m \times q}$ is the steering matrix, $\boldsymbol{n}_k(t) \in \mathbb{C}^m$ is the measurement noise that is considered to be white Gaussian with variance σ^2. The measurements model for the whole array is

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t), \tag{2}$$

where $\boldsymbol{x}(t) = [\boldsymbol{x}_1^T(t), \ldots, \boldsymbol{x}_p^T(t)]^T$, $\boldsymbol{A} = [\boldsymbol{A}_1^T, \ldots, \boldsymbol{A}_p^T]^T$ and $\boldsymbol{n}(t) = [\boldsymbol{n}_1^T(t), \ldots, \boldsymbol{n}_p^T(t)]^T$.

The shift invariance property of each subarray can be expressed by grouping the antennas of each subarray into two groups as depicted in Figure 1. Observe that by shifting the first group by displacement d we obtain the second group. Let us define two selection matrices, $\overline{\boldsymbol{J}} = [\boldsymbol{I}_{m-1},\ \boldsymbol{0}_{m-1}]$ and $\underline{\boldsymbol{J}} = [\boldsymbol{0}_{m-1},\ \boldsymbol{I}_{m-1}]$, where $\boldsymbol{I}_{m-1}$ is the $m-1 \times m-1$ identity matrix, and $\boldsymbol{0}_{m-1}$ is a $m-1 \times 1$ vector containing zero elements. Then the shift invariance property implies that [12]

$$\overline{\boldsymbol{A}}_k \boldsymbol{\Phi} = \overline{\boldsymbol{J}} \boldsymbol{A}_k \boldsymbol{\Phi} = \underline{\boldsymbol{J}} \boldsymbol{A}_k = \underline{\boldsymbol{A}}_k, \tag{3}$$

where $\boldsymbol{\Phi} = \text{diag}\{e^{-j2\pi d \sin(\theta_1)}, \ldots, e^{-j2\pi d \sin(\theta_q)}\}$ is the delay matrix.

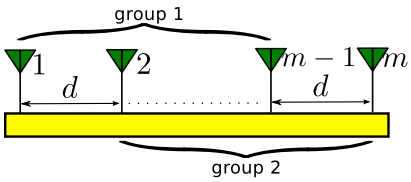

Fig. 1. shift invariance grouping in one subarray

3. THE ESPRIT ALGORITHM

In the conventional array processing scenario, it is assumed that the snapshots of all subarrays are available at a central computation node, where the ESPRIT algorithm [5] is carried out. The eigendecomposition of the covariance matrix $\boldsymbol{R}_{xx} = E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)]$ can be written as

$$\boldsymbol{R}_{xx} = \boldsymbol{E}_{\text{s}} \boldsymbol{\Lambda}_{\text{s}} \boldsymbol{E}_{\text{s}}^H + \boldsymbol{E}_{\text{n}} \boldsymbol{\Lambda}_{\text{n}} \boldsymbol{E}_{\text{n}}^H, \tag{4}$$

[1] Our algorithm can be extended to any shift invariance subarray structure.

where $\boldsymbol{\Lambda}_{\text{s}} \in \mathbb{R}^{q \times q}$ and $\boldsymbol{\Lambda}_{\text{n}} \in \mathbb{R}^{(mp-q) \times (mp-q)}$ are diagonal matrices contain the eigenvalues of the signal and noise subspaces, respectively, $\boldsymbol{E}_{\text{s}} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_q] \in \mathbb{C}^{mp \times q}$ and $\boldsymbol{E}_{\text{n}} = [\boldsymbol{e}_{q+1}, \ldots, \boldsymbol{e}_{mp}] \in \mathbb{C}^{(mp-q) \times q}$ contain the signal and noise subspaces, $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{mp}$ are the eigenvectors of the matrix $\boldsymbol{R}_{xx}$ corresponding to the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_{mp}$. We further partition the ith eigenvector as

$$\boldsymbol{e}_i = [\boldsymbol{e}_{i,1}^T, \ldots, \boldsymbol{e}_{i,p}^T]^T, \tag{5}$$

where $\boldsymbol{e}_{i,k} \in \mathbb{C}^m$ corresponds to the kth subarray.

We define two signal subspaces corresponding to the two groups of sensors introduced earlier as follows

$$\overline{\boldsymbol{E}}_{\text{s}} = (\boldsymbol{I}_p \otimes \overline{\boldsymbol{J}}) \boldsymbol{E}_{\text{s}}, \quad \underline{\boldsymbol{E}}_{\text{s}} = (\boldsymbol{I}_p \otimes \underline{\boldsymbol{J}}) \boldsymbol{E}_{\text{s}}, \tag{6}$$

where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix. Let

$$\boldsymbol{\Psi} = (\overline{\boldsymbol{E}}_{\text{s}}^H \overline{\boldsymbol{E}}_{\text{s}})^{-1} \overline{\boldsymbol{E}}_{\text{s}}^H \underline{\boldsymbol{E}}_{\text{s}}. \tag{7}$$

Then according to [12], property (3) implies that the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are similar in the noise-free case, which means they have identical eigenvalues. Consequently,

$$\theta_i = \sin^{-1}(\arg(\psi_i)/(2\pi d)), \tag{8}$$

where $\psi_i, i = 1, \ldots, q$ are the eigenvalues of the matrix $\boldsymbol{\Psi}$.

In practice, the true covariance matrix is unavailable. However, its sample estimate can be calculated using

$$\hat{\boldsymbol{R}}_{xx} = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{x}(t) \boldsymbol{x}^H(t), \tag{9}$$

where N is the number of available snapshots of the array output. We define $\hat{\boldsymbol{E}}_{\text{s}}, \hat{\overline{\boldsymbol{E}}}_{\text{s}}, \hat{\underline{\boldsymbol{E}}}_{\text{s}}, \hat{\boldsymbol{E}}_{\text{n}}, \hat{\boldsymbol{e}}_i, \hat{\boldsymbol{e}}_{i,k}, \hat{\boldsymbol{\Psi}}$ as the estimates of $\boldsymbol{E}_{\text{s}}, \overline{\boldsymbol{E}}_{\text{s}}, \underline{\boldsymbol{E}}_{\text{s}}, \boldsymbol{E}_{\text{n}}, \boldsymbol{e}_i, \boldsymbol{e}_{i,k}, \boldsymbol{\Psi}$ based on (9), respectively.

4. DECENTRALIZED ESPRIT

In this section, we show how to calculate $\hat{\boldsymbol{E}}_{\text{s}}$ in decentralized fashion without requiring explicit calculation of the matrix $\hat{\boldsymbol{R}}_{xx}$. Then, we suggest a scheme based also on averaging consensus to enable each subarray to access the matrix $\hat{\boldsymbol{\Psi}}$, consequently it can estimate the DoAs of the q sources.

4.1. Decentralized Eigendecomposition

The decentralized eigendecomposition using the power method is proposed in [9], here we adapt this method to our special array structure.

Using the power method [13, p. 330] to calculate $\hat{\boldsymbol{e}}_1$ iteratively, we can write at the $(n+1)$th iteration

$$\hat{\boldsymbol{e}}_1(n+1) = \hat{\boldsymbol{R}}_{xx} \hat{\boldsymbol{e}}_1(n). \tag{10}$$

Substituting (9) into (10) we find

$$\begin{aligned}\hat{\boldsymbol{e}}_1(n+1) &= \left(\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{x}(t)\boldsymbol{x}(t)^H\right)\hat{\boldsymbol{e}}_1(n)\\ &= \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{x}(t)a_t(n),\end{aligned} \tag{11}$$

where $a_t(n) = \boldsymbol{x}(t)^H\hat{\boldsymbol{e}}_1(n)$. If the N scalars $a_t(n), t = 1,\ldots,N$ are available for every subarray, then each subarray can calculate m entry of the eigenvector $\hat{\boldsymbol{e}}_1(n+1)$, these entries are denoted as the vector $\hat{\boldsymbol{e}}_{1,k}(n+1)$, see Equation (5). From (11) it is obvious that $\hat{\boldsymbol{e}}_{1,k}(n+1)$ can be calculated locally at the kth subarray using the following equation

$$\hat{\boldsymbol{e}}_{1,k}(n+1) = \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{x}_k(t)a_t(n), \tag{12}$$

Now we rewrite the scalar $a_t(n)$ as

$$\begin{aligned}a_t(n) = \boldsymbol{x}(t)^H\hat{\boldsymbol{e}}_1(n) &= \sum_{k=1}^{p}\boldsymbol{x}_k^H(t)\hat{\boldsymbol{e}}_{1,k}(n)\\ &= p\left(\frac{1}{p}\sum_{k=1}^{p}a_{t,k}(n)\right) = p\underset{1\le k\le p}{\mathrm{AC}}(a_{t,k}(n)),\end{aligned} \tag{13}$$

where $a_{t,k}(n) = \boldsymbol{x}_k^H(t)\hat{\boldsymbol{e}}_{1,k}(n)$. Each subarray k has access at each iteration to $\hat{\boldsymbol{e}}_{1,k}(n)$ so it can calculate $a_{t,k}(n)$ locally. Then all the subarrays carry out an averaging consensus operation to calculate $a_t(n)$, we denote this Averaging Consensus operation as $\underset{1\le k\le p}{\mathrm{AC}}(a_{t,k}(n))$. In Section 5, we revise the averaging consensus algorithm that can be used to compute (13) in a fully distributed iterative manner by only applying local communication between the neighboring subarrays.

After sufficient number of iterations I_{pm} (where pm stands for power method) the estimated vector $\hat{\boldsymbol{e}}_1(I_{\mathrm{pm}})$ converges to the eigenvector corresponding to the largest eigenvalue [13]. It is further required to normalize this vector using $\hat{\boldsymbol{e}}_1 = \hat{\boldsymbol{e}}_1(I_{\mathrm{pm}})/\left\|\hat{\boldsymbol{e}}_1(I_{\mathrm{pm}})\right\|$. The normalization factor can be calculated in the context of the averaging consensus algorithm, hence

$$\begin{aligned}\left\|\hat{\boldsymbol{e}}_1(I_{\mathrm{pm}})\right\|^2 &= \sum_{k=1}^{p}\hat{\boldsymbol{e}}_{1,k}^H(I_{\mathrm{pm}})\,\hat{\boldsymbol{e}}_{1,k}(I_{\mathrm{pm}})\\ &= p\underset{1\le k\le p}{\mathrm{AC}}(\hat{\boldsymbol{e}}_{1,k}^H(I_{\mathrm{pm}})\,\hat{\boldsymbol{e}}_{1,k}(I_{\mathrm{pm}})).\end{aligned} \tag{14}$$

Let the matrix $\hat{\boldsymbol{U}}_{j-1} = [\hat{\boldsymbol{e}}_1,\ldots,\hat{\boldsymbol{e}}_{j-1}] \in \mathbb{C}^{pm\times(j-1)}$ be the concatenation of $j-1$ estimated eigenvectors. The jth eigenvector is estimated using the following iteration [9]

$$\begin{aligned}\hat{\boldsymbol{e}}_j(n+1) &= (\boldsymbol{I} - \hat{\boldsymbol{U}}_{j-1}\hat{\boldsymbol{U}}_{j-1}^H)\hat{\boldsymbol{R}}_{xx}\hat{\boldsymbol{e}}_j(n)\\ &= \hat{\boldsymbol{e}}'_j(n) - \hat{\boldsymbol{U}}_{j-1}\hat{\boldsymbol{U}}_{j-1}^H\hat{\boldsymbol{e}}'_j(n),\end{aligned} \tag{15}$$

where $\hat{\boldsymbol{e}}'_j(n) = \hat{\boldsymbol{R}}_{xx}\hat{\boldsymbol{e}}_j(n)$. $\hat{\boldsymbol{e}}'_j(n)$ can be calculated similar to $\hat{\boldsymbol{e}}_1(n)$. The term $\hat{\boldsymbol{U}}_{j-1}\hat{\boldsymbol{U}}_{j-1}^H\hat{\boldsymbol{e}}'_j(n)$ can be rewritten as

$$\begin{aligned}\hat{\boldsymbol{U}}_{j-1}\hat{\boldsymbol{U}}_{j-1}^H\hat{\boldsymbol{e}}'_j(n) &= \sum_{i=1}^{j-1}\hat{\boldsymbol{e}}_i\left(\hat{\boldsymbol{e}}_i^H\hat{\boldsymbol{e}}'_j(n)\right)\\ &= \sum_{i=1}^{j-1}\hat{\boldsymbol{e}}_i\left(p\underset{1\le k\le p}{\mathrm{AC}}\left(\hat{\boldsymbol{e}}_{i,k}^H\hat{\boldsymbol{e}}'_{j,k}(n)\right)\right).\end{aligned} \tag{16}$$

Thus, the calculation of $\hat{\boldsymbol{U}}_{j-1}\hat{\boldsymbol{U}}_{j-1}^H\hat{\boldsymbol{e}}'_j(n)$ requires additional $(j-1)$ averaging consensus operations [9].

Note that for the first eigenvector each iteration of the power method required N parallel averaging consensus operations, and one averaging consensus for the normalization. For the jth eigenvector $(N+j-1)$ parallel averaging consensus operations are required at each iteration, and one averaging consensus for the normalization. Then, to calculate the q eigenvectors using the decentralized power method the required consensus operations are

$$N_{\mathrm{eig}} = \sum_{i=1}^{q}\Big(I_{\mathrm{pm}}(N+i-1)+1\Big) = O(qI_{\mathrm{pm}}N). \tag{17}$$

We want to emphasis that at the end of this decomposition each subarray k stores $\hat{\boldsymbol{e}}_{1,k},\ldots,\hat{\boldsymbol{e}}_{q,k}$.

4.2. Decentralized ESPRIT

In this section we show how the matrix $\hat{\boldsymbol{\Psi}}$ can be calculated using averaging consensus.

Multiplying both sides of equation (7) by $\hat{\overline{\boldsymbol{E}}}_{\mathrm{s}}^H\hat{\overline{\boldsymbol{E}}}_{\mathrm{s}}$, we find

$$(\hat{\overline{\boldsymbol{E}}}_{\mathrm{s}}^H\hat{\overline{\boldsymbol{E}}}_{\mathrm{s}})\hat{\boldsymbol{\Psi}} = \hat{\overline{\boldsymbol{E}}}_{\mathrm{s}}^H\hat{\underline{\boldsymbol{E}}}_{\mathrm{s}} \tag{18}$$

Let $\hat{\boldsymbol{B}} = (\hat{\overline{\boldsymbol{E}}}_{\mathrm{s}}^H\hat{\overline{\boldsymbol{E}}}_{\mathrm{s}})$ and $\hat{\boldsymbol{W}} = \hat{\overline{\boldsymbol{E}}}_{\mathrm{s}}^H\hat{\underline{\boldsymbol{E}}}_{\mathrm{s}}$, then (18) becomes

$$\hat{\boldsymbol{B}}\hat{\boldsymbol{\Psi}} = \hat{\boldsymbol{W}}. \tag{19}$$

Each entry of the matrix $\hat{\boldsymbol{B}}$ can be written as

$$\begin{aligned}[\hat{\boldsymbol{B}}]_{i,j} &= \sum_{k=1}^{p}(\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{i,k})^H(\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{j,k})\\ &= p\underset{1\le k\le p}{\mathrm{AC}}((\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{i,k})^H(\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{j,k})).\end{aligned} \tag{20}$$

Similar as in the previous section each subarray locally computes the scalar $(\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{i,k})^H(\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{j,k})$. Then averaging consensus is carried out such that all subarrays learn about the entry $[\hat{\boldsymbol{B}}]_{i,j}$. Thus, using q^2 parallel averaging consensus operations (one for each element in the matrix $\hat{\boldsymbol{B}}$) all the subarrays will have access to $\hat{\boldsymbol{B}}$. Similar as above entries of $\hat{\boldsymbol{W}}$ are given as

$$\begin{aligned}[\hat{\boldsymbol{W}}]_{i,j} &= \sum_{k=1}^{p}(\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{i,k})^H(\underline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{j,k})\\ &= p\underset{1\le k\le p}{\mathrm{AC}}((\overline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{i,k})^H(\underline{\boldsymbol{J}}\hat{\boldsymbol{e}}_{j,k})),\end{aligned} \tag{21}$$

which can be calculated using averaging consensus.

Thus, all the subarrays have local access to the entries of the matrices $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{B}}$. Therefore, Equation (19) can be solved locally in each subarray to obtain estimates of $\hat{\boldsymbol{\Psi}}$. Finally, in each subarray DoA estimates can be computed according to (8). We refer to this algorithm as the decentralized ESPRIT (d-ESPRIT).

The following three important properties of the d-ESPRIT algorithm should be emphasised. i) First, d-ESPRIT is a fully distributed algorithm, where sensors communicate only to compute averaging consensus. The total communication cost of this algorithm is determined by $O(qI_{\rm pm}N)$, which is independent from the number of subarrays. Consequently, adding new subarrays does not increase the communication load at the individual subarrays, which is the case in the centralized ESPRIT where multi-hop networks are involved. ii) Second, d-ESPRIT follows the same steps as conventional ESPRIT, which means that d-ESPRIT is able to estimate the directions of up to $(m-1)p$ sources, while each subarray alone can estimate only the directions of $(m-1)$ sources. iii) Third, the estimation of the signal subspace which is carried out in the first step of d-ESPRIT can be easily replaced by subspace tracking methods while keeping the estimation of the matrix $\boldsymbol{\Psi}$ as explained above. In other words, our algorithm can be easily extended to use the decentralized subspace tracking methods which are introduced in [10] and [11] instead of using patch processing.

5. AVERAGING CONSENSUS

The d-ESPRIT algorithm uses averaging consensus in Equations (13, 14, 16, 20, 21) to calculate the average of p scalar values, where each value is stored on different subarray. In this section, we summarize the iterative averaging consensus algorithm that have been used in the simulation in Section 6 to achieve this calculations. This algorithm is presented in full details in [14].

It is assumed that each subarray can communicate with a set of subarrays that are located in its vicinity. This set is called the neighboring set, and for the kth subarray we denote it as $\mathcal{N}_k$. We also denote the values that we want to average as $\boldsymbol{x}(0) = [x_1(0), \ldots, x_p(0)]^T \in \mathbb{C}^p$, where the state $x_k(0)$ is stored at the kth subarray. For simplicity we assume at each iteration (n) that each subarray k broadcasts its current state $x_k(n)$, and receives the states of all its neighbors, $\{x_i(n)\}_{i\in\mathcal{N}_k}$. Then it updates its state using the following simple iteration

$$\begin{aligned} x_k(n+1) &= x_k(n) + \sum_{i\in\mathcal{N}_k} \alpha_{k,i}\,(x_i(n) - x_k(n)), \\ \alpha_{k,i} &= \frac{1}{\max(|\mathcal{N}_i|, |\mathcal{N}_k|)}, \end{aligned} \tag{22}$$

where $|\mathcal{N}_i|$ is the degree of the ith subarray, which is the cardinality of the set $\mathcal{N}_i$. After sufficient number of iterations $I_{\rm ac}$ (where ac stands for Averaging Consensus), the states $x_k(I_{\rm ac}), k = 1, \ldots, p$ will converge to the average of the initial states $\overline{\boldsymbol{x}(0)} = \frac{1}{p}\sum_{k=1}^{p} x_k(0)$, see [14].

6. SIMULATION RESULTS

We used a network of $p = 6$ subarrays, each of which consists of two antennas separated by half a wavelength. The selection matrices in this case are $\overline{\boldsymbol{J}} = [1,\ 0]$ and $\underline{\boldsymbol{J}} = [0,\ 1]$ The neighboring sets are $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_2 = \{1, 3\}$, $\mathcal{N}_3 = \{1, 2, 4\}$, $\mathcal{N}_4 = \{3, 5, 6\}$, $\mathcal{N}_5 = \{4, 6\}$, and $\mathcal{N}_6 = \{4, 5\}$. Each subarray can communicate only with its neighbors. Two sources are impinging on this network from directions -5 and 5 degrees. The measurements are affected by white Gaussian noise $SNR = -2$ dB. Each subarray collects $N = 100$ snapshots. We compare the Root Mean Square Error (RMSE) calculated over 1000 Monte Carlo runs for three implementations of ESPRIT. i) The first implementation is centralized and uses MATLAB "eig(.)" command[2] to estimate the signal subspace. This implementation is used to benchmark the power method as it does not depend on $I_{\rm pm}$, we call this implementation CB (Centralized Benchmark). ii) The second implementation is also centralized, but it uses the power method to estimate the signal subspace, we refer to this implementation as CPM (Centralized Power Method). iii) The third implementation is the d-ESPRIT algorithm, which is fully distributed implementation that uses the power method and averaging consensus.

In Figure 2, the averaging consensus is run for $I_{\rm ac} = 5$ iterations, and the RMSE is plotted as a function of $I_{\rm pm}$. Note that CB does not depend on $I_{\rm pm}$. However, increasing $I_{\rm pm}$ decreases the RMSE in the other two implementations. For $I_{\rm pm} > 4$, it is clear that CPM and d-ESPRIT have converged, and the RMSE of d-ESPRIT is larger because the consensus averaging needs more than $I_{\rm ac} = 5$ iterations to converge. This will become clearer after we explain the second simulation.

In Figure 3, the number of the power method iterations is fixed to $I_{\rm pm} = 4$, and the RMSE is plotted as a function of $I_{\rm ac}$. Note that the first two implementations CB and CPM do not depend on $I_{\rm ac}$ since they are both centralized. For CPM, the RMSE is still a little bit larger than CB because of the limited number of the power method iterations. Following d-ESPRIT, we can see that its RMSE converges to the RMSE of CPM after $I_{\rm ac} = 6$ iterations. If large $I_{\rm pm}$ and $I_{\rm ac}$ are used, then d-ESPRIT will have the same performance as CB, which is the best performance expected from d-ESPRIT, however, the communication cost will increase linearly with both iterations $I_{\rm pm}$ and I_{av}. Thus, d-ESPRIT offers an additional

[2]For a Hermitian matrix $\boldsymbol{A}$, MATLAB uses the LAPACK function ZHEEV to compute the eigendecomposition of $\boldsymbol{A}$. ZHEEV first reduces $\boldsymbol{A}$ to real tridiagonal form, using unitary similarity transformations, and then the QR algorithm is applied to the tridiagonal matrix to compute the eigenvalues and (optionally) the eigenvectors, for more details see [15] and [13].

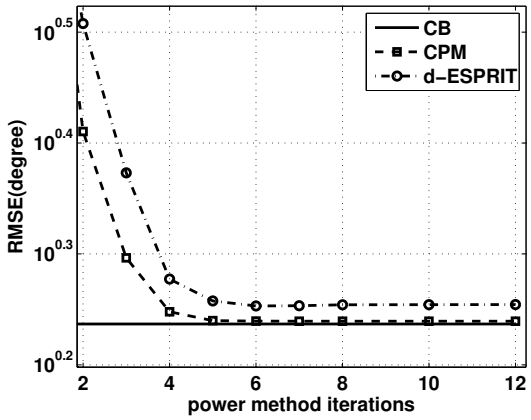

Fig. 2. RMSE as a function of the power method iterations.

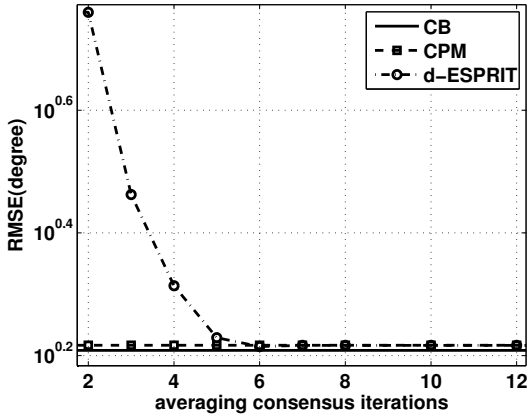

Fig. 3. RMSE as a function of averaging consensus iterations.

advantage, which is a compromise between communication cost and RMSE.

7. CONCLUSIONS

We introduced the d-ESPRIT algorithm for DoA estimation in partly calibrated arrays. This algorithm combines the advantages of the conventional ESPRIT algorithm and averaging consensus algorithm to achieve fully distributed DoA estimation with computation and communication cost independent from the total number of subarrays. However, the computation and communication cost of d-ESPRIT is still dependent on the number of sources, and an ideal implementation which avoids this drawback is yet to be achieved.

References

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 4, pp. 67–94, 1996.

[2] R. Schmidt, "Multiple emitter location and signal parameter estimation," *Antennas and Propagation, IEEE Transactions on*, vol. 34, no. 3, pp. 276–280, 1986.

[3] P. Stoica and K. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 7, pp. 1132–1143, 1990.

[4] M. Viberg, B. Ottersten, and T. Kailath, "Detection and estimation in sensor arrays using weighted subspace fitting," *Signal Processing, IEEE Transactions on*, vol. 39, no. 11, pp. 2436–2449, 1991.

[5] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 7, pp. 984–995, 1989.

[6] M. Pesavento, A. Gershman, and K. Wong, "Direction finding in partly calibrated sensor arrays composed of multiple subarrays," *Signal Processing, IEEE Transactions on*, vol. 50, no. 9, pp. 2103–2115, 2002.

[7] A. Swindlehurst, B. Ottersten, R. Roy, and T. Kailath, "Multiple invariance esprit," *Signal Processing, IEEE Transactions on*, vol. 40, no. 4, pp. 867–881, 1992.

[8] A. Swindlehurst, P. Stoica, and M. Jansson, "Exploiting arrays with multiple invariances using music and mode," *Signal Processing, IEEE Transactions on*, vol. 49, no. 11, pp. 2511–2521, 2001.

[9] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*. IEEE, 2008, pp. 1722–1726.

[10] C. Reyes, T. Hilaire, and C. Mecklenbräuker, "Distributed projection approximation subspace tracking based on consensus propagation," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*. IEEE, 2009, pp. 340–343.

[11] L. Li, A. Scaglione, and J. Manton, "Distributed principal subspace estimation in wireless sensor networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 725–738, 2011.

[12] G. Xu, S. Silverstein, R. Roy, and T. Kailath, "Beamspace esprit," *Signal Processing, IEEE Transactions on*, vol. 42, no. 2, pp. 349–356, 1994.

[13] G.H. Golub and C.F. Van Loan, *Matrix computations*, vol. 3, Johns Hopkins University Press, 1996.

[14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[15] E Anderson, Z Bai, C Bischof, et al., "Lapack users' guide, release 2.0," *SIAM, Philadelphia*, vol. 326, pp. 327, 1995.