Fast Distributed Consensus-based Estimation (Fast-DiCE) for Cooperative Networks

Guang Xu, Henning Paul, Dirk Wübben, Armin Dekorsy
Department of Communications Engineering
University of Bremen, 28359 Bremen, Germany
Email: {xu, paul, wuebben, dekorsy}@ant.uni-bremen.de

***Abstract*—In distributed networks several nodes aim to estimate the signals broadcasted by the sources in a cooperative fashion by exchanging information among neighboring nodes. Consensus based estimation is a specific class of In-Network-Processing (INP) techniques and the Distributed Consensus-based Estimation (DiCE) algorithm is an efficient realization for the Least Squares (LS) estimation problem. In this paper, we modify the update functions of the DiCE algorithm by applying Nesterov's optimal gradient descend method leading to the novel fast-DiCE algorithm. Furthermore, we extend the distributed estimation algorithms with respect to the MMSE-criterion. The performance of the discussed schemes will be evaluated considering two different applications: a cooperative sensor network and the uplink of a small-cell mobile communication network. The results indicate the improved convergence of the fast-DiCE algorithm resulting also in a reduced communication overhead among the cooperating nodes.**

I. Introduction

The topic of consensus and cooperation has always been discussed in the applications of cooperative networks [1]. A variety of distributed consensus algorithms has been specifically applied to these cooperative networks, e.g., distributed Wireless Sensor Networks (WSNs) [2], [3] or densely deployed small cells mobile networks [4]. A common scenario for the cooperative network is that a group of nodes (e.g., sensor nodes or base stations) is connected in a certain topology (e.g. ring, mesh or random) via inter-node links. A message is broadcasted from a source point to all these nodes. The estimate of the common message can be obtained locally at each node, but the estimate of the nodes may vary due to the different channel conditions, so a consensus on this broadcasted message is desired throughout the whole network. A possible way to realize this, is that every node forwards its local estimate to a central node, which reconstructs the common message. But considering the system robustness, a more reliable way, avoiding the failure of the central node is that each node only exchanges, e.g., local estimates with its neighbors for a number of iterations, and by doing so, the common message can be recovered among all these nodes cooperatively. Correspondingly, a variety of algorithms has been proposed for this distributed processing in WSN [3], [5]. In [3], some distributed processing has been set up for the consensus estimates with single hop exchange. But considering the heavily directional communication between the nodes, the overhead can be saved by the algorithm DiCE proposed in [2], where each node broadcasts its own local estimates to the neighbors to reduce the dedicated node to node transmission. What we are interested in is a further reduction of the overhead. From a mathematic perspective, this distributed estimation can be viewed as a constrained convex optimization problem. To solve such problems, a method named Alternating Direction Method of Multiplier (ADMM) [6] was adopted by the DiCE algorithm. But the ADMM can be further optimized by an accelerated ADMM algorithm proposed in [7], which is inspired by the optimal gradient descend method proposed by Nesterov [8]. So referring to this accelerated method, we have improved the DiCE algorithm and obtained our novel algorithm fast-DiCE for cooperative communication. In this paper, we will present its performance and also investigate the behavior of both Zero-Forcing (ZF) and Minimum Mean Square Error (MMSE) based distributed estimation algorithms for selected scenarios.

The remainder of this paper is organized as follows. The considered model is introduced in Section II and the considered approaches for signal estimation in the distributed network are discussed in Section III. After recapping centralized as well as local executed linear equalization schemes, the DiCE algorithm is summarized and the modifications for fast-DiCE are introduced. The performance of these approaches are investigated in Section IV for two different applications. The paper is concluded in Section V.

II. System Description

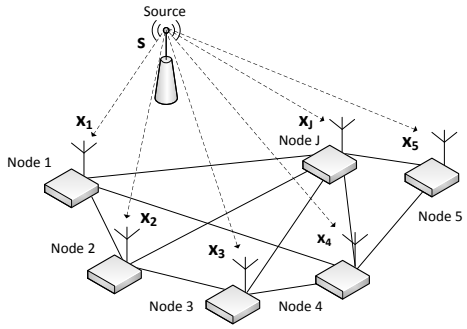

Fig. 1. A network with J nodes is receiving a message $\mathbf{s}$ broadcasted from a common source. Each node exchanges information on $\mathbf{s}$ with its neighboring nodes.

Fig. 1 shows a general scenario where an arbitrary message vector $\mathbf{s}$ is observed by a network of J nodes. This network of nodes is described by a geometric graph $\mathcal{G} := \{\mathcal{J}, \mathcal{E}\}$, in which $\mathcal{J} := \{1, ..., J\}$ denotes the set of nodes and $\mathcal{E}$ represents the set of edges for the linked nodes. For exchanging information, each node $j \in \mathcal{J}$ can only communicate with nodes in its neighborhood $\mathcal{N}_j \subseteq \mathcal{J}$. In this work it is assumed that all inter-node links are ideal and time-invariant. The impact of erroneous links has been investigated in [9].

It is assumed, that the message vector $\mathbf{s}$ is real-valued and contains N elements. At each node j an observation vector $\mathbf{x}_j \in \mathbb{R}^{M\times 1}$ of length M is achieved, which depends linearly on the message $\mathbf{s}$ as indicated by the system equation

$$\mathbf{x}_j = \mathbf{H}_j\mathbf{s} + \mathbf{n}_j \ . \tag{1}$$

Here, the disturbance is indicated by a real-valued matrix $\mathbf{H}_j \in \mathbb{R}^{M\times N}$ and $\mathbf{n}_j$ denotes an additional Gaussian noise term containing elements with variance σ_n^2. Thus, (1) describes a general multiple input multiple output (MIMO) system and can be used to represent numerous applications like multiple antenna point-to-point systems, multiuser scenarios, and sensor networks. Without loss of generality, we assume for the general system model continuous message vectors $\mathbf{s} \in \mathbb{R}^{N\times 1}$. However, discrete values are considered for application scenarios discussed in Section IV as well.

Based on the observation vector $\mathbf{x}_j$ each node can in principle estimate the message vector $\mathbf{s}$ separately. However, when the rank of the disturbance matrix $\mathbf{H}_j$ is smaller than N, this would lead to bad performance. In general, better estimation quality compared to locally estimating the message $\mathbf{s}$ in each node j can be achieved by processing all receive signals $\mathbf{x}_j$, $j \in \mathcal{J}$ jointly. One possible way for joint processing is achieved by forwarding all observations $\mathbf{x}_j$ and all matrices $\mathbf{H}_j$ to a central node and constructing the overall system model

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{2}$$

with the stacked observation vector $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_J^T]^T$, stacked matrix $\mathbf{H} = [\mathbf{H}_1^T, \dots, \mathbf{H}_J^T]^T$, and stacked noise vector $\mathbf{n} = [\mathbf{n}_1^T, \dots, \mathbf{n}_J^T]^T$. The estimate can then be calculated in the central node based on all observations. In order to reduce the required communication overhead for forwarding all local observations to the central node, distributed estimation approaches are of particular interest as presented in the subsequent section.

III. Detection Schemes

A. Linear Equalization

Based on the collected observations (2) the central node can solve the Least Squares (LS) problem

$$\hat{\mathbf{s}} = \arg\ \min_{\mathbf{s}} \|\mathbf{x} - \mathbf{H}\mathbf{s}\|^2 \tag{3}$$

achieving the estimate $\hat{\mathbf{s}}$. The solution for (3) is given by the well-known Zero-Forcing (ZF) linear equalizer

$$\hat{\mathbf{s}}_{\mathrm{ZF}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x} = \mathbf{H}^+\mathbf{x} \tag{4}$$

which filters the observation vector $\mathbf{x}$ with the Moore-Penrose Pseudo-Inverse of the disturbance matrix $\mathbf{H}^+ = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x}$. It is well known, that the ZF equalizer suffers from noise amplification and better results can generally be achieved by applying the Minimum Mean Square Error (MMSE) criterion. The MMSE criterion reduces the overall estimation error and calculates as

$$\hat{\mathbf{s}}_{\mathrm{MMSE}} = (\mathbf{H}^T\mathbf{H} + \sigma_n^2\mathbf{I}_N)^{-1}\mathbf{H}^T\mathbf{x} \tag{5a}$$

$$= (\underline{\mathbf{H}}^T\underline{\mathbf{H}})^{-1}\underline{\mathbf{H}}^T\underline{\mathbf{x}} = \underline{\mathbf{H}}^+\underline{\mathbf{x}} \ , \tag{5b}$$

with the augmented disturbance matrix $\underline{\mathbf{H}} = [\mathbf{H}^T, \sigma_n\mathbf{I}_N]^T$ and the augmented observation vector $\underline{\mathbf{x}} = [\mathbf{x}^T, \mathbf{0}_{1,N}]^T$ [10]. As can be seen in (5b), the MMSE equalization can be written as a ZF solution with respect to the augmented disturbance matrix $\underline{\mathbf{H}}$ and the augmented received signal vector $\underline{\mathbf{x}}$.

In case each node contains at least as much observations as unknown symbols, i.e. $M \geq N$, each node may also perform a local linear estimation with respect to (1). Thus, the local ZF estimate at node j is given by

$$\hat{\mathbf{s}}_{j,\mathrm{ZF}} = (\mathbf{H}_j^T\mathbf{H}_j)^{-1}\mathbf{H}_j^T\mathbf{x}_j = \mathbf{H}_j^+\mathbf{x}_j \ . \tag{6}$$

and the MMSE solution corresponds to $\hat{\mathbf{s}}_{j,\mathrm{MMSE}} = \underline{\mathbf{H}}_j^+\underline{\mathbf{x}}_j$. However, in general the estimates $\hat{\mathbf{s}}_{j,\mathrm{ZF}}$ at the different nodes will usually differ and the performance compared to the central solution is worse.

B. DiCE Algorithm

The DiCE algorithm proposed in [2] solves the LS problem (3) in a distributed fashion by information exchange among neighboring nodes. The distributed processing is performed by decomposing the central estimation problem into local problems with the consensus constraint $\mathbf{s}_i = \mathbf{s}_j$ for $i \in \mathcal{N}_j$, resulting in the consensus-constrained quadratic optimization problem

$$\{\hat{\mathbf{s}}_j | j \in \mathcal{J}\} = \underset{\{\mathbf{s}_j | j\in\mathcal{J}\}}{\arg\min} \sum_{j=1}^{J} \|\mathbf{x}_j - \mathbf{H}_j\mathbf{s}_j\|^2 \tag{7a}$$

$$\text{s.t.} \quad \mathbf{s}_j = \mathbf{s}_i \quad \forall\ j \in \mathcal{J},\ i \in \mathcal{N}_j \ . \tag{7b}$$

According to the constraint (7b), the local estimate $\mathbf{s}_j$ of node j should be in an agreement with the estimates $\mathbf{s}_i$ of its neighboring nodes $i \in \mathcal{N}_j$. However, due to the direct coupling of the estimates $\mathbf{s}_j, j \in \mathcal{J}$, this set of local estimation problems cannot be solved in parallel. In order to realize a distributed calculation of the problem, auxiliary variables $\mathbf{z}_j$ are introduced at each node j for decoupling the local estimates leading to the new constraint equation

$$\text{s.t.} \quad \mathbf{z}_j = \mathbf{s}_j, \mathbf{s}_i = \mathbf{z}_j \quad \forall\ j \in \mathcal{J},\ i \in \mathcal{N}_j \ . \tag{8}$$

In order to solve the underlying constrained optimization problem, the linear distributed estimation is derived with the help of the Augmented Lagrangian method (ALM) [11] and Alternating Direction Method of Multipliers (ADMM) [6],

leading to the update equations for the local variables per iteration at node j [2]

$$\mathbf{s}_j^{k+1} = \left(\mathbf{H}_j^T\mathbf{H}_j + \frac{|\mathcal{N}_j^+|}{\mu}\mathbf{I}_N\right)^{-1} \cdot \left[\mathbf{H}_j^T\mathbf{x}_j + \sum_{i\in\mathcal{N}_j^+}\left(\frac{1}{\mu}\mathbf{z}_i{}^k + \boldsymbol{\lambda}_{ji}^k\right)\right] , \tag{9a}$$

$$\mathbf{z}_j^{k+1} = \frac{\mu}{|\mathcal{N}_j^+|}\sum_{i\in\mathcal{N}_j^+}\left[\frac{1}{\mu}\mathbf{s}_i{}^{k+1} - \boldsymbol{\lambda}_{ij}^k\right] , \tag{9b}$$

$$\boldsymbol{\lambda}_{ij}^{k+1} = \boldsymbol{\lambda}_{ij}^k - \frac{1}{\mu}\left(\mathbf{s}_i^{k+1} - \mathbf{z}_j^{k+1}\right) . \tag{9c}$$

Here, the set $\mathcal{N}_j^+ = \mathcal{N}_j \cup \{j\}$ contains the neighboring nodes of node j and itself. $\mathbf{s}_j^{k+1}$ and $\mathbf{z}_j^{k+1}$ are the estimates at node j after iteration $k+1$ and $\boldsymbol{\lambda}_{ij}^{k+1}$ represent the Lagrangian multiplier at node j.

By initializing the variables $\mathbf{z}_j^0 = \boldsymbol{\lambda}_{ij}^0 = \mathbf{0}$ the first estimate $\mathbf{s}_j^1$ at node j depends only on the local observation $\mathbf{x}_j$ and the disturbance matrix $\mathbf{H}_j$

$$\mathbf{s}_j^1 = \left(\mathbf{H}_j^T\mathbf{H}_j + \frac{|\mathcal{N}_j^+|}{\mu}\mathbf{I}_N\right)^{-1} \cdot \mathbf{H}_j^T\mathbf{x}_j . \tag{10}$$

After exchanging the local estimates among neighbors, each node j determines the auxiliary variables according to (9b) and shares these estimates again with its neighbors. Finally, the Lagrangian multipliers are updated (9c) to prepare the calculations in the next iteration. Thus, in each iteration the local estimates $\mathbf{s}_j^{k+1}$ and the auxiliary variables $\mathbf{z}_j^{k+1}$ are updated by considering information from the neighboring nodes and the previous own estimates, i.e., taking information from nodes $i \in \mathcal{N}_j^+$. Each update is afterwards broadcasted again to these neighbors.

In contrast, the Lagrangian multiplier $\boldsymbol{\lambda}_{ij}^{k+1}$ could be calculated at node j based on the locally available estimates $\mathbf{s}_i^{k+1}$ and $\mathbf{z}_j^{k+1}$. In a network with error-free inter-node links, the exchange of these variables is not corrupted. Thus, the Lagrangian multipliers do not have to be shared among the neighboring nodes. However, if the exchange of local estimates is not perfect (i.e., error-prone network), the convergence of the DiCE algorithm can only be ensured by exchanging also the Lagrangian multipliers. To this end, node j has to unicast to each of its neighbors $i \in \mathcal{N}_j$ the variable $\boldsymbol{\lambda}_{ij}^{k+1}$ and has to receive the variables $\boldsymbol{\lambda}_{ji}^{k+1}$ [9], [12]. In order to reduce the required overhead for these unicast transmissions, the recently published Reduced Overhead DiCE (RO-DiCE) omits the exchange of Lagrangian multipliers using a approximated update function [13]. After some iterations, all of the local estimates will approximately be the same, and the local estimates approaches the centralized solution for a sufficient number of iterations. Please note, that an optimization of the penalty term μ in (9) is beyond the scope of this paper and is set to the constant value of 1 similar to [2].

In order to improve the estimation performance, the DiCE algorithm can also be extended to the MMSE criterion by considering the augmented disturbance matrix $\underline{\mathbf{H}}_j$ and the augmented observation vectors $\underline{\mathbf{x}}_j$ in (9).

C. Fast-DiCE Algorithm

In each iteration step of DiCE the local variables $\mathbf{s}_j$, $\mathbf{z}_j$ and $\boldsymbol{\lambda}_{ij}$ have to be exchanged among the neighboring nodes leading to a considerable communication overhead. In order to increase the convergence speed and thereby reducing also the communication overhead, we propose here a modification of the DiCE algorithm by adopting Nesterov's optimal gradient descend method [8]. To this end, the underlying ADMM approach is improved by predicting the auxiliary variables and the Lagrangian multipliers as proposed by [7].

The main idea is to calculate at node j a predictor $\tilde{\mathbf{z}}_{ji}^{k+1}$ for the auxiliary variable of node i based on the two latest received estimates $\mathbf{z}_i^{k+1}$ and $\mathbf{z}_i^k$ from node i by

$$\tilde{\mathbf{z}}_{ji}^{k+1} = \mathbf{z}_i^{k+1} + \gamma_{k+1}\left(\mathbf{z}_i^{k+1} - \mathbf{z}_i^k\right) . \tag{11}$$

Thus, the newest estimate $\mathbf{z}_i^{k+1}$ is basically extended by the gradient of the auxiliary variable $\mathbf{z}_i^{k+1} - \mathbf{z}_i^k$ weighted by the step size γ_{k+1}. All predicted auxiliary variables $\tilde{\mathbf{z}}_{ji}^{k+1}$, $i \in \mathcal{N}_j$, are then used for the next update of the local estimate $\mathbf{s}_j^{k+2}$ (12a). Similarly, predictors $\boldsymbol{\lambda}_{ji}^k$ for the Lagrangian multipliers are calculated in the same way and used in the update equations for $\mathbf{s}_j^{k+2}$ and $\mathbf{z}_j^{k+2}$ in (12a) and (12b), respectively. Thus, the update equations for the novel fast-DiCE algorithm are given by

$$\mathbf{s}_j^{k+1} = \left(\mathbf{H}_j^T\mathbf{H}_j + \frac{|\mathcal{N}_j^+|}{\mu}\mathbf{I}_N\right)^{-1} \cdot \left[\mathbf{H}_j^T\mathbf{x}_j + \sum_{i\in\mathcal{N}_j^+}\left(\frac{1}{\mu}\tilde{\mathbf{z}}_{ji}^k + \tilde{\boldsymbol{\lambda}}_{ji}^k\right)\right] , \tag{12a}$$

$$\mathbf{z}_j^{k+1} = \frac{\mu}{|\mathcal{N}_j^+|}\sum_{i\in\mathcal{N}_j^+}\left[\frac{1}{\mu}\mathbf{s}_i{}^{k+1} - \tilde{\boldsymbol{\lambda}}_{ij}^k\right] , \tag{12b}$$

$$\boldsymbol{\lambda}_{ij}^{k+1} = \tilde{\boldsymbol{\lambda}}_{ij}^{\,k} - \frac{1}{\mu}\left(\mathbf{s}_i^{k+1} - \mathbf{z}_j^{k+1}\right) , \tag{12c}$$

with the calculation of the predictors

$$\tilde{\mathbf{z}}_{ji}^{k+1} = \mathbf{z}_i^{k+1} + \gamma_{k+1}\left(\mathbf{z}_i^{k+1} - \mathbf{z}_i^k\right) , \tag{13a}$$

$$\tilde{\boldsymbol{\lambda}}_{ji}^{\,k+1} = \boldsymbol{\lambda}_{ji}^{k+1} + \gamma_{k+1}\left(\boldsymbol{\lambda}_{ji}^{k+1} - \boldsymbol{\lambda}_{ji}^k\right) \tag{13b}$$

and step size parameter in iteration $k+1$ [8]

$$\gamma_{k+1} = \frac{\alpha_k - 1}{\alpha_{k+1}} \text{ and } \alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2} . \tag{14}$$

By initializing $\alpha_0 = 1$ the step size γ_k takes the values listed in Tab. I and is increasing from 0 up to 0.5. As the difference between two consecutive estimates of the auxiliary variables and the Lagrangian multipliers will be rather large, the step size is chosen to be small. With an increasing number of

iterations, the difference will be reduced, so the step size is increased to still achieve a prediction gain. As shown by the authors in [7] the convergence rate of the ADMM can be increased from $\mathcal{O}(k)$ to $\mathcal{O}(k^2)$ by this Nesterov-type modification.

TABLE I
PARAMETER α_k AND STEP SIZE γ_k IN STEP k

k	1	2	3	4	5	8
α_k	1.725	1.912	1.971	1.990	1.997	1.999
γ_k	0	0.379	0.463	0.488	0.496	0.499

By initializing the predictors as $\tilde{\mathbf{z}}_{ji}^0 = \tilde{\boldsymbol{\lambda}}_{ji}^0 = \mathbf{0}$ the fast-DiCE algorithm achieves the same estimates $\mathbf{s}_j^1$, auxiliary variables $\mathbf{z}_j^1$ and Lagrangian multipliers $\boldsymbol{\lambda}_{ij}^1$ at node j as the DiCE algorithm (9) in the first iteration. Afterwards, the predictors are updated according to (13) leading to $\tilde{\mathbf{z}}_{ji}^1 = \mathbf{z}_i^1$ and $\tilde{\boldsymbol{\lambda}}_{ji}^1 = \boldsymbol{\lambda}_{ji}^1$ due to the step size $\gamma_1 = 0$. Correspondingly, also the updates in the second iteration lead to the same results as the DiCE algorithm. However, for $k \geq 2$ the step size γ_k is non vanishing such that the predictors $\tilde{\mathbf{z}}_{ij}$ and $\tilde{\boldsymbol{\lambda}}_{ji}$ from now on depend on the last two latest received estimates for the auxiliary variables and the Lagrangian multipliers.

Like the DiCE algorithm, once the local estimate $\mathbf{s}_j^{k+1}$, the auxiliary variable $\mathbf{z}_j^{k+1}$, and the multipliers $\boldsymbol{\lambda}_{ij}^{k+1}$ at node j are updated according to (12), they are delivered to its neighboring nodes $i \in \mathcal{N}_j$. Based on these exchanged variables the predictors $\tilde{\mathbf{z}}_{ji}^{k+1}$ and $\tilde{\boldsymbol{\lambda}}_{ji}^{k+1}$ are then updated at every node $j \in \mathcal{J}$ following (13) improving the estimation of the local variables in the next iteration step.

It should be noted, that the predictors $\tilde{\mathbf{z}}_{ji}$ and $\tilde{\boldsymbol{\lambda}}_{ji}$ are calculated locally in each node j and do not have to be exchanged among the nodes. Thus, the communication overhead per iteration remains the same for fast-DiCE and DiCE. Nevertheless, as demonstrated in the subsequent section, the fast-DiCE algorithm requires fewer iterations to achieve the same estimation quality as the DiCE algorithm. Correspondingly, the overall communication effort as well as the latency is reduced by this novel algorithm. However, for the fast-DiCE each node j needs $2 \cdot |\mathcal{N}_j^+|$ additional buffers for storing the auxiliary variables and Lagrangian multipliers from the last iteration and complexity per iteration is slightly increased.

IV. PERFORMANCE EVALUATION

In the following, the performance of the proposed fast-DiCE algorithm will be compared to the DiCE algorithm by investigating the estimation error and the bit error rate (BER) with respect to the number of iterations for three different scenarios. Furthermore, both approaches will be implemented with respect to the ZF- and the MMSE-criterion and will be compared also to the corresponding central and local linear estimators.

A. Scenarios

In scenario A, a cooperative sensor network as depicted in Fig. 2 is considered. Here, $J = 6$ sensor nodes each equipped with $n_R = 4$ antennas monitor one common source with $n_T = 2$ transmit antennas resulting in a MIMO system with $N = n_R = 2$ input variables and $M = n_R = 4$ output variables per node j. For each link between the source and sensor j the channel coefficients are assumed to be real valued and zero-mean normal distributed with the variance σ_j^2 representing the individual channel gains

$$\left[\sigma_j^2\right] = [0.8\ 0.15\ 0.5\ 1\ 0.2\ 0.7]\ . \tag{15}$$

Here, the message broadcasted from the source is assumed to be zero mean, normal distributed with variance $\sigma_s^2 = 1$. A fully connected sensor network without link errors is assumed such that all sensors exchange erroneously messages with each other.

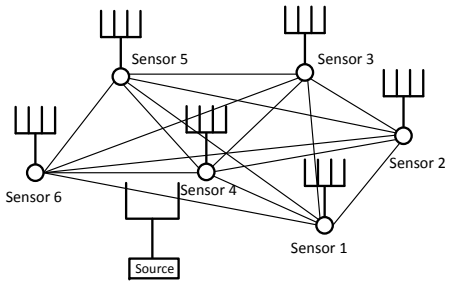

Fig. 2. Scenario A: Fully connected wireless sensor network where $J = 6$ sensors with $n_R = 4$ receive antennas monitoring a source containing $n_T = 2$ transmit antennas.

Scenario B considers an uplink scenario of a mobile communication network with $J = 3$ small cells (SC), each equipped with $n_R = 4$ receive antennas as shown in Fig. 3. The $U = 4$ users transmit uncoded BSPK messages simultaneously on the same frequency band by their $n_T = 2$ antennas. Thus, the source vector $\mathbf{s} \in \mathcal{A}^N$ comprises the $N = U \cdot n_T$ transmit symbols of all users and $\left[\mathbf{H}_j^{(u-1)\cdot n_T+1} \ldots \mathbf{H}_j^{u \cdot n_T}\right]$ denotes the channel coefficients between the receive antennas at small cell j and the n_T transmit antennas of user u. Obviously, the local system equation (1) is underdetermined, whereas the central system (2) achieved by collecting all receive signal in a central node remains overdetermined. In this scenario, the aim of the cooperating network of small cells is to perform joint multiuser detection in a distributed fashion. In the case of $J = 3$ SCs the full meshed backhaul network reduces to a simple ring topology.

It is assumed, that the channel coefficients are again zero-mean normal distributed with variance σ_{uj}^2 indicating the channel gain between user u and small cell j. For scenario B we assume

$$\left[\sigma_{uj}^2\right] = \begin{bmatrix} 1 & 0.5 & 0.3 & 0.1 \\ 0.6 & 1 & 0.8 & 0.6 \\ 0.1 & 0.5 & 0.8 & 1 \end{bmatrix}\ . \tag{16}$$

Scenario C equals scenario B but assumes channels with equal gain, i.e., $\left[\sigma_{uj}^2\right] = \mathbf{1}_{3,4}$.

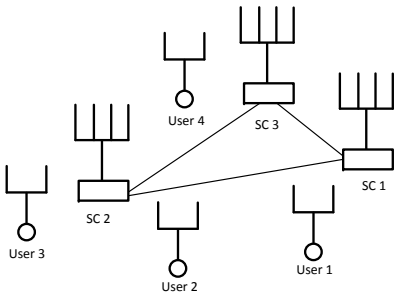

Fig. 3. Scenario B &, C: $U = 4$ users each with $n_T = 2$ transmit antennas send messages simultaneously to $J = 3$ SCs in the uplink, where each SC has $n_R = 4$ receive antennas and all SCs are connected in a ring with each other.

B. Estimation and Convergence Performance

In order to investigate the convergence behavior of the fast-DiCE and the DiCE algorithm, the estimation performance after iteration k is evaluated by means of the average Mean Squared Error (aMSE). The aMSE is defined as the square error between the local estimates $\mathbf{s}_j^k$ and the message $\mathbf{s}$ averaged over all receiving nodes j and realizations

$$\text{aMSE}(k) = \frac{1}{J}\sum_{j=1}^{J} \mathrm{E}\{\|\mathbf{s} - \mathbf{s}_j^k\|^2\} \,. \tag{17}$$

The simulation will be conducted by means of the Monte Carlo method, i.e., for each scenario, a large number of random realizations of the message vectors $\mathbf{s}$, the channel matrix $\mathbf{H}_j$ and the noise vectors $\mathbf{n}_j$ are considered. The ZF- and MMSE-based fast-DiCE and DiCE algorithms are compared with the central and local linear estimators based on the ZF and the MMSE criterion. For the central estimation, the (17) simplifies to $\text{aMSE} = \mathrm{E}\{\|\mathbf{s} - \hat{\mathbf{s}}\|^2\}$.

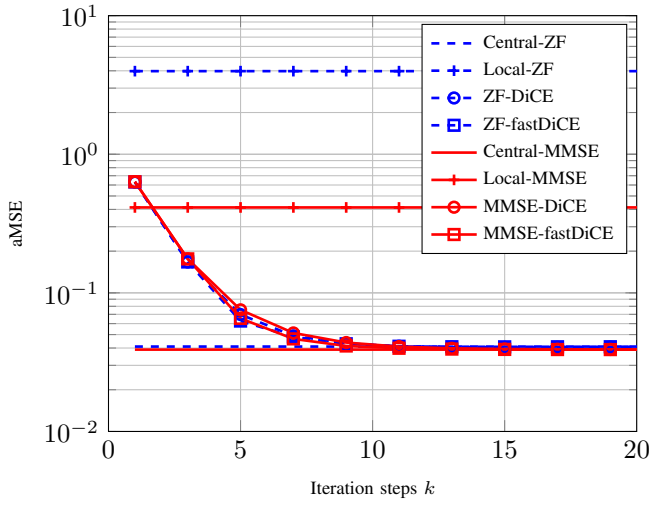

Fig. 4. Averaged Mean Square Error (aMSE) for Scenario A at SNR $= 5$ dB

Fig. 4 shows the performance of all approaches for scenario A and SNR $= 5$ dB. Compared to the local estimators, the error is significantly reduced by the centralized solutions. As the overall system is strong overdetermined ($N = 2$ input and $M \cdot J = 24$ output variables), almost no difference between the central ZF and MMSE solution can be observed. The aMSE of the DiCE and the fast-DiCE implementations in the first iteration is slightly higher compared to the local MMSE solution but converges to the centralized solution after some iterations. The fast-DiCE implementation shows an improved convergence behavior compared to DiCE approach.

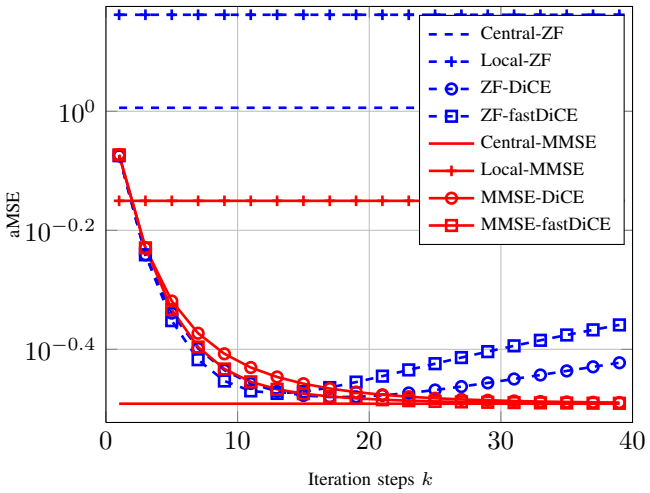

Fig. 5. Averaged Mean Square Error (aMSE) for Scenario B at SNR $= 5$ dB

For Scenario B the aMSE of all approaches is shown in Fig. 5. As each local estimation problem is underdetermined ($N = U \cdot n_T = 4 \cdot 2 = 8$ input and $M = 4$ output variables), the local ZF and MMSE estimators can not achieve sufficient results. By collecting all receive signals in central node, an overdetermined system with $M \cdot J = 4 \cdot 3 = 12$ output variables is obtained leading to promising estimates by the centralized solutions, and the MMSE criterion leads to significant improvements compared to the ZF solution. Overall, the fast-DiCE shows a better performance than DiCE algorithm for both ZF and MMSE criterion. Additionally, it can be seen in the figure that the aMSE curves of both ZF-based distributed algorithms approach the central MMSE solution and then return back to the central ZF solution after certain iterations, and they drop even faster than the MMSE-based algorithms within the first few iterations, which needs to be further investigated.

Furthermore, we have investigated the convergence of fast-DiCE and DiCE in another way by taking the gradient of the Lagrangian cost function regarding the objective function (7a) and the constraint (8) is as a stopping criterion [9], which can indicate the convergence behavior of the distributed algorithms. The simulation will be terminated when the stopping criterion falls below a threshold, and accordingly the total number of iterations for each simulation will be counted. The value for the threshold should not be set too low otherwise both fast-DiCE and DiCE will take many iterations to reach the threshold and the difference becomes not obvious.

Regarding the convergence performance, Fig. 6 and Fig. 7

depict the cumulative density function (CDF) of the required number of iterations for all distributed algorithms applied in scenario A and B, respectively. It can be observed that in both scenarios, the fast-DiCE always has a better performance than DiCE. In general, fast-DiCE needs fewer iterations to converge. But when we compare both MMSE- or ZF-based

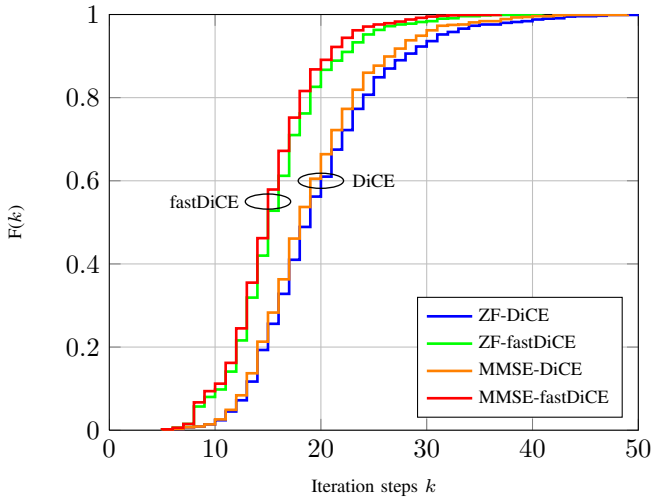

Fig. 6. Empirical cumulative density function (CDF) of required number of iterations for terminating the simulation over 1000 random realizations for Scenario A, at SNR = 5 dB

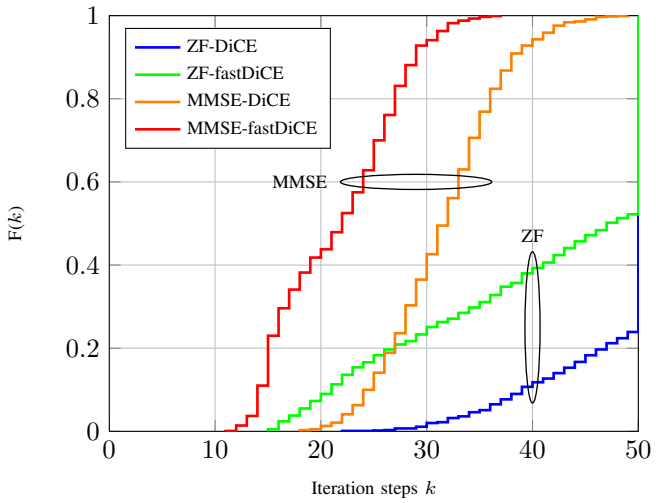

Fig. 7. Empirical cumulative density function of required number of iterations for terminating the simulation over 1000 random realizations Scenario B, at SNR = 5 dB

estimation, approximately the same performance can be found in scenario A at SNR of 5 dB. As shown in Fig. 6, the CDF of MMSE-based distributed algorithms is only slightly faster than the ZF-based algorithms, which perfectly matches with the aMSE curves shown in Fig. 4. But in scenario B at the same SNR, the MMSE-based algorithm outperforms the ZF-based algorithm as the CDF shown in Fig. 7, where all of MMSE-based simulations can be finished within 50 iterations while the ZF-based needs a lot more iterations to terminate the simulation for most runs, which is also corresponding to the behavior of aMSE shown in Fig. 5. The CDF can only show the convergence behavior of the algorithm, but it does not reflect the estimation quality between MMSE and ZF-based algorithms due to simple stopping criterion.

C. Bit Error Rate Performance

With regard to the bit error performance of all the algorithms in scenario B and C. The uncoded BPSK messages transmitted by the users are locally estimated and exchanged among the SCs. After some number of k iterations according to the distributed estimation algorithms, the local estimate $\mathbf{s}_j$ is then obtained by a hard decision at SC j. The bit errors are then counted and averaged over all the SCs and realizations. As a comparison, the bit errors for central ZF and MMSE estimation are also counted.

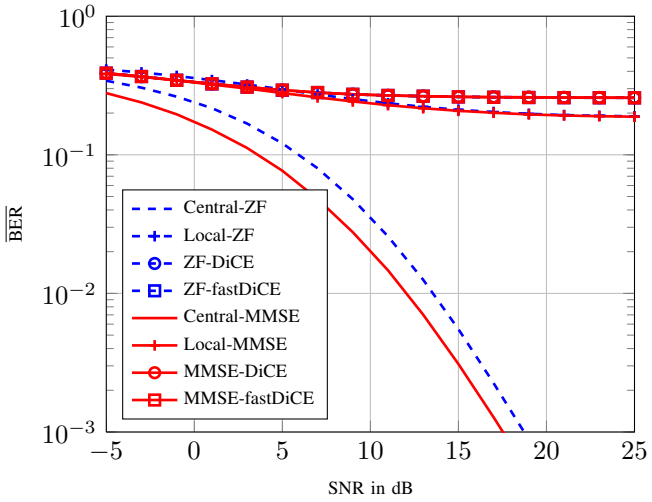

Fig. 8. Simulated averaged bit error rate (BER) of the algorithms for the Scenario B at the first iteration

Fig. 8 depicts the BER performance of all algorithms at the first iteration for the scenario B. It can be clearly seen that a large gain is achieved by the central estimation compared to the local estimation. Then look at the performance of the distributed estimation algorithms, they are similar at the first iteration. But when compared to the local-ZF they just achieve a small gain at low SNRs and lose the gain at high SNRs. At the second iteration as shown in Fig. 9, still no big difference can be found among the distributed algorithms, but they start to converge to the central MMSE solution according to the aMSE curve shown above, and their performance has much improved compared to the local estimation. After several iterations, the performance of fast-DiCE will be better than DiCE as can be seen in Fig. 10 for a fixed number of 30 iterations. In this figure, the fast-DiCE has a much lower error floor than DiCE due to its faster convergence. The error floor is formed because of the insufficient number of iterations for the distributed algorithms to converge at high SNRs (due to a more accurate central solution). Thus less communication effort is required

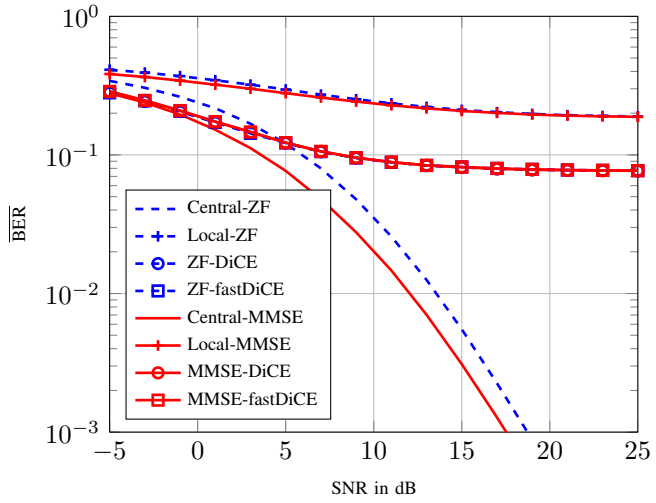

Fig. 9. Simulated averaged bit error rate (BER) of the algorithms for the Scenario B at the second iteration

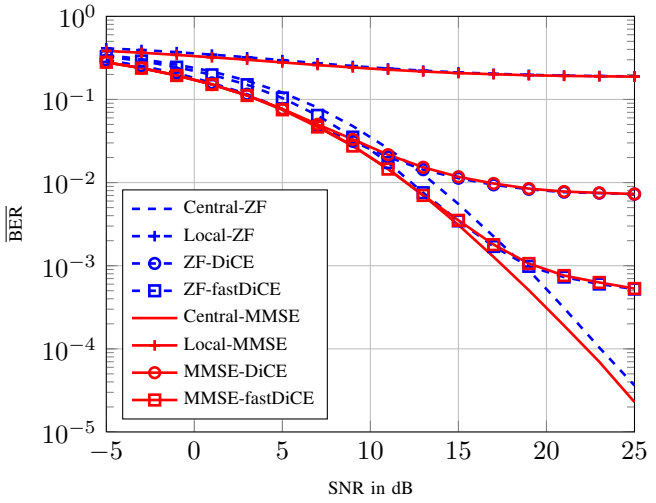

Fig. 10. Simulated averaged bit error rate (BER) of the algorithms for the Scenario B at $k = 30$ iterations

between SCs by using fast-DiCE compared to DiCE in order to achieve the same performance. But with increasing of SNR, approximately from 13 dB, the BER curve of MMSE-fast-DiCE and ZF-fast-DiCE start to merge together while the central MMSE and ZF still apart, since both MMSE- and ZF-based algorithms behave similarly at first few iterations as approaching to the central MMSE solution, therefore their BER curves are overlapped in the high SNR range when the number of iterations is not sufficient. As a comparison, we also look at the performance of all algorithms for the scenario C at the $k = 30$ iteration, which is shown in Fig. 11. It can be seen that the performance of central estimation as well as the distributed estimation are better than the results shown in Fig. 10 due to the good channel conditions, and because of that, the error floors are lower, since fewer iterations are needed for the distributed algorithm to converge. Besides, it can still be found that the MMSE-fast-DiCE outperforms ZF-fast-DiCE at low SNRs and has a better performance than MMSE/ZF-DiCE at high SNRs.

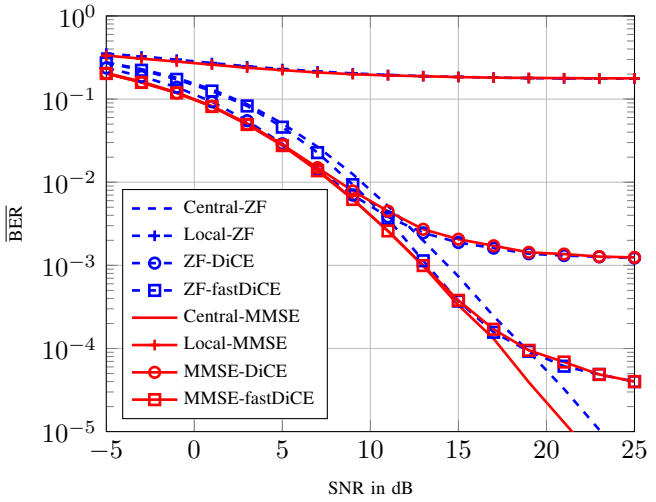

Fig. 11. Simulated averaged bit error rate (BER) of the algorithmsfor the Scenario C at $k = 30$ iterations

V. Conclusion

In this paper, we presented an improved algorithm named fast-DiCE that outperforms the previous DiCE algorithm for In-Network-Processing. In this algorithm, two "predictors" are introduced to optimize the update of the estimates in order to accelerate the convergence of the distributed estimation, and the overall communication overhead can be reduced throughout all the receiving nodes without losing accuracy. Both algorithms with respect to the MMSE and ZF equalization have been investigated in two different example applications. We showed that the fast-DiCE always has a better performance than the DiCE, and the MMSE-based estimation shows a faster convergence than ZF-based algorithms in general, but a good stopping criterion needs to be investigated, which will be done in future work. Besides, the fast-DiCE algorithm will be investigated in an error-prone network where the inter-node links are noisy and sparsely disconnected. The convergence rate of fast-DiCE also needs to be investigated mathematically, and the performance of fast-DiCE may be possibly further optimized.

VI. Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 317941. The authors would like to acknowledge the contributions of their colleagues in iJOIN, although the views expressed are those of the authors and do not necessarily represent the project.

References

[1] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems,," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[2] H. Paul, J. Fliege, and A. Dekorsy, "In-Network-Processing: Distributed Consensus-Based Linear Estimation," *Communications Letters, IEEE*, vol. 17, no. 1, pp. 59–62, Jan. 2013.
[3] H. Zhu, A. Cano, and G. Ginanakis, "Distributed Consensus-based Demodulation: Algorithms and Error Analysis," *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 2044–2054, Jun. 2010.
[4] H. Paul, B.-S. Shin, D. Wübben, and A. Dekorsy, "In-Network-Processing for Small Cell Cooperation in Dense Networks," in *IEEE VTC2013-Fall Workshop (CLEEN 2013)*, Las Vegas, USA, Sept. 2013.
[5] I. D. Schizas, G. Mateos, and G. Ginanakis, "Distributed LMS for Consensus-Based In-Network Adaptive Processing," *IEEE Trans. Signal Processing*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.
[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multiplier," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
[7] T. Goldstein, B. O'Donoghue, and S. Setzer, "Fast Alternating Direction Optimization Methods," in *CAM report 12-35*, UCLA, USA, May 2012.
[8] Y. Nesterov, "A Method of Solving a Convex Programming Problem with Convergence Rate $O(1/k^2)$," *Soviet. Math. Dokl.*, vol. 27, 1983.
[9] H. Paul, B.-S. Shin, D. Wübben, and A. Dekorsy, "Distributed Consensus-based linear estimation with erroneous links," in *ITG Workshop on Smart Antennas (WSA) 2013*, Stuttgart, Germany, Mar. 2013.
[10] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE Extension of V-BLAST based on Sorted QR Decomposition," in *IEEE 58th Semiannual Vehicular Technology Conference (VTC2003-Fall)*, Orlando, FL, USA, Oct. 2003.
[11] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Science+Business Media, 2006.
[12] B.-S. Shin, "Fehlertolerante verteilte Schätzalgorithmen für drahtlose Netzwerke (in german)," in *Diploma Thesis*, University of Bremen, 2013.
[13] B.-S. Shin, H. Paul, D. Wübben, and A. Dekorsy, "Reduced Overhead Distributed Consensus-Based Estimation Algorithm," in *International Workshop on Cloud-Processing in Heterogeneous Mobile Communication Networks (IWCPM) 2013*, Atlanta ,USA, Dec. 2013.